

WordPerfect[®]

VERSION 6.0

WINDOWS

*M*ACROS MANUAL

WordPerfect and WPCorp are registered trademarks of WordPerfect Corporation within the United States and other countries. WordPerfect Office is a trademark of WordPerfect Corporation worldwide. All other brand and product names are trademarks or registered trademarks of their respective companies.

Version 6.0
©WordPerfect Corporation 1994
All Rights Reserved.
Printed in U.S.A.

MNUSWWP60X1A—4/94
21 —74 —74

WordPerfect Corporation • 1555 N. Technology Way • Orem, Utah 84057-2399 U.S.A.
Telephone: (801) 225-5000 • Telex: 820618 • Fax: (801) 222-5077

Table of Contents

Theory and Practice	Introduction	3
	Macro Facility	4
	Macro Language Concepts	6
	Variables, Constants, Operators, and Expressions	10
	Macro Control Statements	25
Creating Macros	Recording Macros	35
	Writing Macros	38
	Dialog Boxes	43
	Macro Examples	48
	Macro Menus	50
	Creating Callback Functions	54
	Dynamic Data Exchange	65
Assigning a Macro	69	
Programming Commands Reference	// (Comment)	73
	APPACTIVATE	73
	APPEXECUTE	74
	APPEXECUTEEXT	74
	APPLICATION	76
	APPLOCATE	77
	ASSERT	77
	ASSIGN	79
	BEEP	80
	BIFFILEPATH	81
	BIFINFO	81
	BIFREAD	83
	BIFWRITE	85
	BREAK	88
	CALL	89
	CANCEL	90
	CASE	92
	CASE CALL	93
	CHAIN	94
	CHARLEN	96
	CHARPOS	97
	CTON	98
	DDEEXECUTE	99
	DDEEXECUTEEXT	100
	DDEINITIATE	101
	DDEPOKE	102
	DDEREQUEST	102
	DDETERMINATE	103
	DDETERMINATEALL	104
DECLARE	104	
DEFAULTUNITS	106	
DIALOGADDCHECKBOX	107	
DIALOGADDCOLORWHEEL	108	
DIALOGADDCOMBOBOX	111	
DIALOGADDCONTROL	113	
DIALOGADDCOUNTER	115	
DIALOGADDEDITBOX	118	
DIALOGADDFILENAMEBOX	120	
DIALOGADDFRAME	121	
DIALOGADDGROUPBOX	123	
DIALOGADDHLINE	125	
DIALOGADDHOTSPOT	126	
DIALOGADDICON	127	
DIALOGADDLISTBOX	129	
DIALOGADDLISTITEM	131	
DIALOGADDPUPBUTTON	132	
DIALOGADDPUSHBUTTON	134	
DIALOGADDRADIOBUTTON	136	
DIALOGADDSCROLLBAR	137	
DIALOGADDTEXT	140	
DIALOGADDVIEWER	142	
DIALOGADDVLINE	143	
DIALOGDEFINE	145	
DIALOGDESTROY	147	
DIALOGDISPLAY	148	
DIALOGHANDLE	152	
DIALOGUNDISPLAY	153	
DISCARD	154	
DLLCALL	155	
DLLFREE	157	
DLLLOAD	158	
ENDAPP	160	

ERROR	160	ONCANCEL CALL	199
EXISTS	161	ONDDEADVISE CALL	200
FOR-ENDFOR	163	ONERROR	201
FOREACH-ENDFOR	164	ONERROR CALL	202
FORNEXT-ENDFOR	166	ONNOTFOUND	203
FRACTION	167	ONNOTFOUND CALL	204
FUNCTION	168	PAUSE	205
GETNUMBER	171	PERSIST	206
GETSTRING	172	PERSISTALL	207
GETUNITS	173	PROCEDURE	208
GLOBAL	174	PROMPT	210
GO	175	QUIT	211
IF-ELSE-ENDIF	176	REPEAT-UNTIL	212
IFPLATFORM-ENDIFPLATFORM	178	RETURN	213
INDIRECT	179	RUN	215
INTEGER	181	SENDKEYS WAIT	216
LABEL	182	SPEED	219
LOCAL	183	STRLEN	220
MENU	184	STRNUM	222
MENULIST	185	STRPOS	223
MESSAGEBOX	186	STRUNIT	224
MMSPLAY	189	SUBCHAR	226
MMSPEAK	190	SUBSTR	227
MMSPEAKCLIPBOARD	191	SWITCH-ENDSWITCH	228
MMSTOPSPEECH	192	TOLOWER	230
NEST	193	TOUPPER	231
NEWDEFAULT	193	UNITSTR	232
NEXT	194	USE	234
NOTFOUND	195	VARERRCHK	236
NOTOC	196	WAIT	236
NUMSTR	197	WHILE-ENDWHILE	237
ONCANCEL	198		

Product Commands Reference

AbbreviationCreate	241	AttributeAppearanceOff	246
AbbreviationDelete	241	AttributeAppearanceOn	247
AbbreviationDlg	242	AttributeAppearanceToggle	248
AbbreviationExpand	242	AttributeNormal	249
AboutDlg	243	AttributePosition	249
Advance	243	AttributePositionToggle	250
AdvanceDlg	244	AttributeRelativeSize	250
Append	244	AttributeRelativeSizeToggle	251
AppMaximize	244	AttributeSizeRatios	251
AppMinimize	244	BackTab	252
AppMove	245	Backup	252
AppRestore	245	BackupMinutes	253
AppSize	246	BackupOriginalDoc	253
AppTaskList	246	BarcodeDlg	254
AssociateDlg	246	BarcodePOSTNET	254

BaselinePlacement	255	BoxContentType	290
Binding	255	BoxCounter	290
BindingOptionsDlg	256	BoxCreate	291
BlockProtect	256	BoxDelete	292
BoldKey	257	BoxEdit	293
BookmarkBlock	257	BoxEditByCounter	293
BookmarkCreate	257	BoxEditNext	294
BookmarkDelete	258	BoxEditPrevious	295
BookmarkDlg	258	BoxEnd	295
BookmarkFind	259	BoxEquationColor	296
BookmarkMove	259	BoxEquationDefaultFont	297
BookmarkRename	260	BoxEquationFilename	298
BorderBottomLine	260	BoxEquationFont	298
BorderColor	261	BoxEquationFontSize	301
BorderCornerRadius	262	BoxEquationRedisplay	301
BorderDropShadow	263	BoxEquationSaveAsGraphic	302
BorderFillColors	264	BoxFill	303
BorderInsideSpacing	265	BoxHeight	303
BorderLeftLine	266	BoxHorizontalAlignment	304
BorderOutsideSpacing	267	BoxHorizontalPosition	305
BorderRightLine	268	BoxImageBlackWhiteThreshold	305
BorderSeparatorLine	269	BoxImageBrightness	306
BorderSetSpacing	270	BoxImageContrast	307
BorderStyleCopy	270	BoxImageFill	307
BorderStyleCreate	272	BoxImageFlipX	308
BorderStyleDelete	272	BoxImageFlipY	309
BorderStyleEdit	273	BoxImageInvertColors	309
BorderStyleEnd	274	BoxImageMonochrome	310
BorderStyleName	275	BoxImageMove	310
BorderStyleRetrieve	275	BoxImageRetrieve	311
BorderStyleSave	276	BoxImageRotate	312
BorderTopLine	276	BoxImageSave	312
BorderUseBorderColor	277	BoxImageScaling	313
BoxAttachTo	278	BoxImageShowBackground	313
BoxBorder	279	BoxInitialStyleDlg	314
BoxBorderDlg	279	BoxOLEEditObject	314
BoxCaptionEdit	279	BoxOverlap	315
BoxCaptionEditEnd	280	BoxPosDlg	315
BoxCaptionFormatting	280	BoxResetBorder	316
BoxCaptionInitialStyle	281	BoxResetCaption	316
BoxCaptionNumberStyle	283	BoxResetContent	316
BoxCaptionPosition	284	BoxResetContentRender	317
BoxCaptionRotation	286	BoxResetCounter	317
BoxChangeLineHeight	286	BoxResetFill	318
BoxChangeStyle	287	BoxResetPosition	318
BoxContentEdit	288	BoxResetTextFlow	318
BoxContentPosition	288	BoxStyleCopy	319
BoxContentPreserveAspectRatio	289	BoxStyleCreate	320

BoxStyleDelete	320	CloseNoSave	343
BoxStyleEdit	321	CodePage	344
BoxStyleEnd	322	CodePageDlg	345
BoxStyleName	322	ColorAdd	345
BoxStyleRetrieve	323	ColorDelete	346
BoxStyleSave	323	ColorModify	347
BoxTextAngle	324	ColorPaletteCreate	347
BoxTextDefaultStyle	325	ColorPaletteDelete	348
BoxTextEdit	327	ColorPaletteDlg	348
BoxTextEditEnd	327	ColorPaletteRename	349
BoxTextFilename	327	ColorPaletteSelect	349
BoxTextFlow	328	ColumnBorderDlg	350
BoxTextFlowContour	329	ColumnDefineDlg	350
BoxTextFlowSquare	329	ColumnsDefinition	350
BoxUpdateDisplay	329	ColumnsOff	351
BoxVerticalAlignment	330	CommentConvert	351
BoxVerticalPosition	330	CommentCreate	352
BoxWidth	331	CommentEdit	352
BoxWrapDlg	332	CommentInformation	352
BulletDlg	332	CommentInsertTime	352
ButtonBarCreateDlg	332	CommentInsertUserInitials	353
ButtonBarEditDlg	332	CommentInsertUserName	353
ButtonBarOptionsDlg	333	ConditionalEndOfPage	353
ButtonBarSelect	333	ConvertCaseInitialCaps	354
ButtonBarSetFont	333	ConvertCaseLowercase	354
ButtonBarSetLocation	334	ConvertCaseUppercase	354
ButtonBarSetRows	334	Copy	355
ButtonBarSetStyle	335	CounterCreate	355
ButtonBarShow	336	CounterDecrement	356
ButtonBarShowToggle	336	CounterDelete	356
CaseToggle	336	CounterDisplay	357
Center	337	CounterDlg	358
CenterCurrentPage	337	CounterEditLevel	358
CenterPageDlg	337	CounterEditLevelMethod	359
CenterPages	338	CounterEditLevelName	360
ChapterNumber	338	CounterEditMethod	360
ChapterNumberDecrement	339	CounterEditName	361
ChapterNumberDisplay	339	CounterIncrement	362
ChapterNumberIncrement	339	CounterSetMethod	362
ChapterNumberMethod	340	CounterSetValue	363
Char	340	CrossRefMarkReference	364
CharMappingDlg	341	CrossRefMarkTarget	365
ClearDoc	341	CustomBoxDlg	365
ClipboardAppendTo	341	DateCode	365
ClipboardRetrieve	342	DateFormat	366
ClipboardSaveTo	342	DateFormatDlg	366
Close	342	DateText	367
CloseGraphicsControlBar	343	DDEChangeUpdateMode	367

DDECreateLink	367	EditPaste	391
DDEDeleteLink	368	EndCenterOrAlignment	392
DDEEditLink	368	EndnoteCreate	392
DDEPasteLink	369	EndnoteEdit	393
DDEUpdateLink	369	EndnoteEditDlg	393
DecimalAlignmentCharacter	370	EndnoteMinimum	393
DelayCodes	370	EndnoteNewNumber	394
DelayCodesDlg	371	EndnoteNumberDecrement	394
DeleteAppend	371	EndnoteNumberDisplay	395
DeleteCharNext	371	EndnoteNumberIncrement	395
DeleteCharPrevious	371	EndnoteNumberMethod	395
DeleteToBeginningOfWord	372	EndnoteOptions	396
DeleteToEndOfLine	372	EndnoteOptionsDlg	397
DeleteToEndOfPage	372	EndnotePlacement	397
DeleteToEndOfWord	373	EndnotePlacementDlg	398
DeleteWord	373	EndnoteSetNumberDlg	398
DirectoryCreate	373	EndnotesSpaceBetween	398
Display	374	EnterKey	399
DisplayZoom	374	EnvelopeDlg	399
DocCompare	375	EquationBaseFontDlg	399
DocCompareAddMarkingsDlg	375	EquationCreate	399
DocCompareRemoveMarkingsDlg	376	EquationPalette	400
DocCompareRemoveRedline	376	EquationPaletteToggle	400
DocInitialFont	376	EquationRetrieveDlg	400
DocInitialFontDlg	379	EquationZoomDlg	401
DocMaximize	380	ExpandDlg	401
DocMinimize	380	FigurePaletteDlg	401
DocMove	380	FigurePaletteDlgEnd	401
DocNext	381	FigureRetrieveDlg	402
DocPrevious	381	FileChangeAttributes	402
DocRedlineMethodDlg	381	FileChangeDir	402
DocRestore	381	FileCopy	403
DocSize	382	FileDelete	403
DocSummaryConfig	382	FileInsert	404
DocSummaryDelete	384	FileInsertDlg	405
DocSummaryDlg	384	FileMove	405
DocSummaryPrint	384	FileNew	405
DocSummarySaveAs	385	FileOpen	406
DocSumSetFull	386	FileOpenDlg	407
DotLeader	388	FileRemoveDirectory	408
DoubleSidedPrint	389	FileSave	408
DraftZoom100	389	FileSaveAsDlg	410
DraftZoomMargin	390	FillColors	410
DragTextBegin	390	FillSetGradient	411
DragTextEnd	390	FillSetWPG2	412
EditComment	391	FillStyleCopy	414
EditCopy	391	FillStyleCreate	415
EditCut	391	FillStyleDelete	415

FillStyleEdit	416	FootnoteEditDlg	441
FillStyleEnd	417	FootnoteMinimum	442
FillStyleName	418	FootnoteNewNumber	442
FillStyleRetrieve	418	FootnoteNewNumberDlg	443
FillStyleSave	419	FootnoteNumberDecrement	443
FindDlg	419	FootnoteNumberDisplay	443
FindFiles	420	FootnoteNumberIncrement	443
FirstLineIndent	421	FootnoteOptions	444
FloatingCellBegin	421	FootnoteOptionsDlg	445
FloatingCellCreate	422	FootnoteSeparatorLine	445
FloatingCellEdit	422	FootnoteSpaceBetween	447
FloatingCellEnd	422	ForcePage	447
FloatingCellFormula	423	ForcePageDlg	448
FloatingCellNumAlignCurrency	423	FormatMarginsDlg	448
FloatingCellNumberDateFormat	424	GenerateDoc	448
FloatingCellNumberDecDigits	425	GenerateDoc	448
FloatingCellNumberFormat	425	GetAcceleratorText	449
FloatingCellNumberNegNumber	426	GetData	450
FloatingCellNumberRound	426	GraphicBoxStyleDlg	452
FloatingCellNumberUseCommas	427	GraphicCreateBox	452
FloatingCellNumberUseCurrency	428	GraphicCreateChart	452
FloatingCellNumSelectCurrency	428	GraphicCreateTextArt	453
FlushRight	430	GraphicCustomMenuDlg	453
Font	430	GraphicsBoxCaptionDlg	453
FontDlg	433	GraphicsBoxContentDlg	453
FontDoubleUnderlineToggle	433	GraphicsBoxFindDlg	454
FontExtraLargeToggle	433	GraphicsBoxSizeDlg	454
FontFineToggle	434	GraphicsFigureEditAllDlg	454
FontHiddenToggle	434	GraphicsLineColor	454
FontItalicToggle	434	GraphicsLineCreate	455
FontLargeToggle	435	GraphicsLineDelete	456
FontOutlineToggle	435	GraphicsLineEdit	456
FontRedlineToggle	435	GraphicsLineEditNext	457
FontShadowToggle	435	GraphicsLineEditPrevious	457
FontSize	436	GraphicsLineEnd	458
FontSmallCapsToggle	436	GraphicsLineHorizontalPosition	458
FontSmallToggle	436	GraphicsLineLength	459
FontStrikeoutToggle	437	GraphicsLineSpacing	460
FontSubscriptToggle	437	GraphicsLineStyle	461
FontSuperscriptToggle	437	GraphicsLineThickness	461
FontUnderlineToggle	438	GraphicsLineType	462
FontVeryLargeToggle	438	GraphicsLineUpdateDisplay	463
FooterA	438	GraphicsLineUseColor	463
FooterB	439	GraphicsLineVerticalPosition	464
FooterSeparationDistance	440	GraphicsStylesDlg	464
FootnoteContinuedMessage	440	HardColumnBreak	465
FootnoteCreate	441	HardPageBreak	465
FootnoteEdit	441	HardPageBreakInsert	465

HardReturn	466	ImportOptionsDlg	483
HardReturnInsert	466	ImportSetAsciiEncap	483
HardSpace	466	ImportSetAsciiFieldDelimiter	484
HeaderA	466	ImportSetAsciiRecordDelimiter	484
HeaderB	467	ImportSetAsciiStrip	485
HeaderFooterDistanceDlg	468	ImportSetDestination	485
HeaderFooterDlg	468	ImportSetDictPath	486
HeaderFooterPlacement	468	ImportSetFields	486
HeaderFooterPlacementDlg	469	ImportSetFileName	487
HeaderSeparationDistance	469	ImportSetFilePath	487
HelpCoachDlg	470	ImportSetIncludeNames	488
HelpContents	470	ImportSetPanel	488
HelpContextSensitive	470	ImportSetPassword	489
HelpHowDoI	471	ImportSetQuery	489
HelpHowToUseHelp	471	ImportSetRange	490
HelpKeystrokes	471	ImportSetServer	491
HelpMacroManual	471	ImportSetSource	491
HelpSearchForHelpOn	472	ImportSetSQLQuery	492
HelpTutorial	472	ImportSetUserID	492
HelpWhatIs	472	ImportUpdateDlg	493
HiddenText	473	Indent	493
HiddenTextShowAll	473	IndentLeftRight	493
HideBars	474	IndexDefineDlg	493
HLineCreate	474	IndexDefinition	494
Hypertext	474	IndexMark	496
HypertextControlBar	475	InhibitInput	496
HypertextCreate	475	InitialCodesStyleDlg	497
HypertextCreateDlg	476	InsertDefaultBullet	497
HypertextDelete	476	InsertFilename	497
HypertextEdit	476	InsertFilenameWithPath	498
HypertextEditDlg	477	InsertOtherCodesDlg	498
HypertextExecute	477	InvokeDialog	498
HypertextNext	478	Justification	499
HypertextPrevious	478	JustifyAll	499
HypertextReturnFrom	478	JustifyCenter	500
Hyphen	479	JustifyFull	500
Hyphenation	479	JustifyLeft	500
HyphenationIgnore	479	JustifyRight	500
HyphenationSoftReturn	480	KeepTextTogetherDlg	501
HyphenationZoneLeft	480	Kerning	501
HyphenationZoneRight	480	KerningSpacing	502
ImportCreateDlg	481	KeyboardSelect	502
ImportDlg	481	KeyboardSelectDlg	503
ImportDoEdit	481	LabelsDefine	503
ImportDoImport	482	LabelsDelete	504
ImportDoLink	482	LabelsDlg	505
ImportDoUpdate	482	LabelsFileCreate	505
ImportEditDlg	483	LabelsFileDescription	506

LabelsFileSelect	506	ListMark	535
LabelsOff	507	MacroAssignDlg	536
LabelsSelect	507	MacroCommandInserter	536
Language	507	MacroEditControlBar	536
LanguageDlg	508	MacroMenuAdd	537
LastOpenedAdd	508	MacroMenuDelete	538
Leading	508	MacroPause	538
LineCreateDlg	509	MacroStatusPrompt	538
LineEditDlg	509	MailDlg	539
LineHeight	509	MarginAdjustLeft	539
LineHeightDlg	510	MarginAdjustRight	540
LineHyphenationDlg	510	MarginBottom	540
LineNumbering	511	MarginLeft	541
LineNumberingDlg	513	MarginReleaseInsert	541
LineNumberingMethod	513	MarginRight	541
LineNumberSet	514	MarginTop	542
LineSegmentColor	514	MarkCrossRef	542
LineSegmentCreate	515	MarkCrossRefUpdateList	542
LineSegmentDelete	516	MarkIndex	543
LineSegmentEdit	516	MarkIndexUpdateList	543
LineSegmentEnd	517	MarkList	544
LineSegmentSetUser	517	MarkListUpdateList	544
LineSegmentSetUserData	518	MarkTableOfAuthorities	544
LineSegmentSetUserEnd	518	MarkTableOfContents	545
LineSegmentSetWPG2	519	MarkToAUpdateList	545
LineSegmentSpacing	520	MasterDocCondense	545
LineSegmentWidth	520	MasterDocCondenseDlg	546
LineSpacing	521	MasterDocExpand	546
LineSpacingDlg	521	MasterDocSubdocDlg	547
LineStyleCopy	521	MatchExtendSelection	547
LineStyleCreate	522	MatchLimit	547
LineStyleDelete	523	MatchPositionAfter	548
LineStyleEdit	524	MatchPositionBefore	548
LineStyleEnd	525	MatchSelection	548
LineStyleName	525	MatchWithAttributes	548
LineStyleRetrieve	526	MatchWithFont	549
LineStyleSave	526	MatchWithFontSize	552
LinkUpdate	527	MenuAddItem	552
ListDefineDlg	527	MenuDeleteItem	554
ListDefinition	527	MenuSelect	554
ListDefinitionCreate	530	MenuSelectDlg	555
ListDefinitionDelete	530	MergeBlankField	555
ListDefinitionEditBox	531	MergeCode	556
ListDefinitionMode	531	MergeCodesDisplayEdit	558
ListDefinitionRename	532	MergeCodesDisplayRun	558
ListDefinitionRetrieve	532	MergeCodesDlg	559
ListDefinitionStyle	533	MergeCondition	559
ListEditPageFormat	535	MergeDataFile	560

MergeDlg	560	OutlineFamilyPromote	579
MergeEnvelope	560	OutlineFamilyRight	579
MergeEnvelopeDefinitionBegin	561	OutlineFamilySelect	580
MergeEnvelopeDefinitionEnd	561	OutlineFamilyShow	580
MergeFileAssociate	562	OutlineFamilyUp	581
MergeFilesDlg	562	OutlineHideShowingBodyText	581
MergeFileType	562	OutlineNumberDlg	581
MergeFormDlg	563	OutlineOff	581
MergeMark	563	OutlineParagraphCopy	582
MergePageBreak	563	OutlineParagraphDelete	582
MergePrintText	564	OutlineParagraphDemote	582
MergeRepeat	564	OutlineParagraphDragDown	583
MergeRun	565	OutlineParagraphDragUp	583
MergeSelect	566	OutlineParagraphMove	583
MergeTableDataDlg	566	OutlineParagraphNumber	584
MergeTextDataDlg	566	OutlineParagraphPromote	584
MinusSign	567	OutlineParagraphSelect	584
MoveModeEnd	567	OutlineRight	585
NewHFWOccurrence	567	OutlineSetParagraphNumber	585
NumberingOptionDlg	568	OutlineShow	585
OLEAcceptLinks	568	OutlineStyleCopy	586
OLECancelLink	568	OutlineStyleCreate	588
OLECancelLinks	569	OutlineStyleDelete	589
OLEChangeLink	569	OutlineStyleDescription	591
OLEInitializeLinks	569	OutlineStyleEditBegin	591
OLEInsertDlg	570	OutlineStyleEditEnd	593
OLEInsertObject	570	OutlineStyleLevel	594
OLELinksDlg	570	OutlineStyleNumberFormat	595
OLEPasteDlg	571	OutlineStyleNumbersOnly	596
OLESetLinkUpdate	571	OutlineStyleRename	597
OLEUpdateLink	571	OutlineStyleRetrieve	597
OutlineBarSetup	572	OutlineStyleSave	598
OutlineBeginNew	572	OutlineStyleSelect	599
OutlineBodyText	574	OutlineUp	600
OutlineBodyTextToggle	575	OutputOption	601
OutlineChangeToBodyText	575	OutputOptionsSetOffsetJogger	601
OutlineDefineDlg	575	OverstrikeCreate	602
OutlineDown	575	OverstrikeCreateDlg	602
OutlineEnd	576	OverstrikeEdit	603
OutlineFamilyCopy	576	PageBorderDlg	603
OutlineFamilyCut	576	PageNumber	603
OutlineFamilyDemote	577	PageNumberDecrement	604
OutlineFamilyDown	577	PageNumberDisplay	604
OutlineFamilyDragDown	577	PageNumberDisplayFormat	604
OutlineFamilyDragUp	578	PageNumberFormat	605
OutlineFamilyHide	578	PageNumberIncrement	605
OutlineFamilyLeft	578	PageNumberingDlg	605
OutlineFamilyMove	579	PageNumberMethod	606

PageNumberPosition	606	PosPageNext	627
PageNumberSettingsDlg	608	PosPagePrevious	628
PageSuppressDlg	608	PosPageTop	628
PageZoomFull	609	PosParagraphNext	628
PaneNext	609	PosParagraphPrevious	629
PanePrevious	609	PosScreenDown	629
PaperSizeAdd	609	PosScreenLeft	629
PaperSizeDelete	611	PosScreenRight	629
PaperSizeDlg	612	PosScreenUp	630
PaperSizeResave	612	PosSelectBottom	630
PaperSizeSelect	614	PosSelectTop	630
ParagraphBorderDlg	614	PosTableBegin	631
ParagraphFormatDlg	615	PosTableBlockBeg	631
ParagraphHangingIndent	615	PosTableCell	631
ParagraphNumberDefDlg	615	PosTableCellBottom	632
ParagraphSpacing	616	PosTableCellDown	632
PasteLink	616	PosTableCellNext	632
PasteSpecial	617	PosTableCellPrevious	633
PauseCommand	617	PosTableCellTop	633
PauseKey	618	PosTableCellUp	633
PauseSet	618	PosTableColBottom	633
PosCellDown	619	PosTableColumnTop	634
PosCellNext	619	PosTableEnd	634
PosCellPrevious	620	PosTableGoPrevious	634
PosCellUp	620	PosTableRowBegin	635
PosCharacter	620	PosTableRowEnd	635
PosCharNext	621	PosTableTopCellNext	635
PosCharPrevious	621	PosTableTopCellPrevious	636
PosColBottom	621	PosWordNext	636
PosColFirst	622	PosWordPrevious	636
PosColLast	622	PowerBarShow	636
PosColNext	622	PrefBackupDlg	637
PosColPrevious	623	PrefBackupPath	637
PosColTop	623	PrefBeep	638
PosDocBottom	623	PrefButtonBar	639
PosDocTop	623	PrefDisplayDlg	639
PosDocVeryTop	624	PrefDisplaySet	639
PosFunction	624	PrefDocSummary	641
PosGoPrevious	624	PrefDocSummaryDlg	641
PosGoToDlg	625	PrefEnvironment	642
PosLineBeg	625	PrefEnvSettingsDlg	643
PosLineDown	625	PrefHyphenation	643
PosLineEnd	625	PrefImportDlg	644
PosLineUp	626	PrefImportOptions	644
PosLineVeryBeg	626	PrefLiteCodes	645
PosLineVeryEnd	626	PrefLocationDlg	646
PosPage	626	PrefLocationOfDocuments	646
PosPageBottom	627	PrefLocationOfGraphicsFiles	647

PrefLocationOfHyphenationFiles . . .	648	QuickFormatDlg	674
PrefLocationOfMacroFiles	648	QuickFormatToggle	675
PrefLocationOfPrinterFiles	649	QuickmarkFind	675
PrefLocationOfSpreadsheetFiles	650	QuickmarkSet	675
PrefLocationOfTemplates	650	Redisplay	676
PrefMenu	651	RedlineCharacter	676
PrefMergeCodes	652	RedlineMethod	676
PrefMetafileOptions	653	RepeatDlg	677
PrefPowerBar	653	RepeatSet	677
PrefPowerBarEdit	654	RepeatValue	678
PrefPowerBarFontFaceList	654	ReplaceAll	678
PrefPowerBarFontSizeList	655	ReplaceBackward	679
PrefPrintCopies	655	ReplaceCurrent	679
PrefPrintDoc	656	ReplaceDlg	680
PrefPrintRedline	656	ReplaceForward	680
PrefPrintSettingsDlg	657	ReplaceString	681
PrefPrintSizeRatio	657	ReplaceWithAttributes	681
PrefRevealCodes	658	ReplaceWithCase	682
PrefRevealCodesDlg	659	ReplaceWithFont	683
PrefRuler	660	ReplaceWithFontSize	685
PrefSave	661	ReselectLastSelection	686
PrefStatusBar	661	ResetKeyboardMenu	686
PrefStatusBarEdit	662	RevealCodes	687
PrefStatusBarOptions	662	RulerBarShow	687
PrefStatusBarOptionsDlg	663	SaveAll	688
PrefToA	664	ScrollScreenLeft	688
PrefUnitsOfMeasure	664	ScrollScreenRight	688
PrefZoom	665	SearchCaseSensitive	688
PrintCopies	666	SearchFindWholeWordsOnly	689
PrintCopiesBy	666	SearchInSelection	689
PrintDlg	667	SearchNext	690
PrintDocOnDisk	667	SearchPrevious	690
PrinterCommand	668	SearchString	691
PrinterCommandDlg	669	SearchWrap	691
PrinterControlDlg	669	SecondaryPageNumber	692
PrinterInitialize	669	SecondaryPageNumberDecrement	692
PrinterPause	669	SecondaryPageNumberDisplay	693
PrinterSelect	670	SecondaryPageNumberIncrement	693
PrinterSelectDlg	670	SecondaryPageNumberMethod	693
PrintFullDoc	670	SelectAll	694
PrintGraphicsQuality	671	SelectCell	694
PrintInkColor	671	SelectCellDown	694
PrintMultiplePages	672	SelectCellDownArrow	695
PrintPage	673	SelectCellLeft	695
PrintSelection	673	SelectCellRight	696
PrintTextQuality	673	SelectCellUp	696
QuickFinderDlg	674	SelectCellUpArrow	696
QuickFormat	674	SelectCharNext	697

SelectCharPrevious	697	ShowBoxes	714
SelectColumnBottom	697	ShowGraphics	715
SelectColumnNext	698	ShowHiddenText	715
SelectColumnPrevious	698	ShowSymbols	715
SelectColumnTop	698	SoftHyphen	715
SelectDelete	699	SoftSpace	716
SelectDocBottom	699	Sort	716
SelectDocTop	699	SortAction	716
SelectDocVeryTop	700	SortCaseOrder	717
SelectLineBegin	700	SortDlg	717
SelectLineDown	700	SortKeys	718
SelectLineEnd	701	SortSelectString	718
SelectLineUp	701	SortType	719
SelectLineVeryBegin	701	SoundClipCreate	719
SelectLineVeryEnd	702	SoundClipDelete	720
SelectMode	702	SoundClipDlg	721
SelectOff	702	SoundClipPlayNext	721
SelectOn	703	SoundClipSetAttributes	721
SelectPage	703	SoundClipSetDescription	722
SelectPageNext	703	SoundClipSetFilename	722
SelectPagePrevious	704	SpecificDeleteStringBegin	723
SelectParagraph	704	SpecificReplaceStringBegin	723
SelectParagraphNext	704	SpecificSearchStringBegin	724
SelectParagraphPrevious	705	SpecificStringEnd	724
SelectRectangle	705	SpreadsheetShowLinkCode	725
SelectSave	705	SpreadsheetUpdateOnRetrieve	725
SelectScreenDown	706	StatusBarShow	726
SelectScreenLeft	706	StyleCodes	727
SelectScreenRight	707	StyleCopy	727
SelectScreenUp	707	StyleCreate	730
SelectSentence	707	StyleDelete	730
SelectSentenceNext	708	StyleDescription	732
SelectSentencePrevious	708	StyleEditBegin	733
SelectTable	708	StyleEditEnd	735
SelectTableColumn	709	StyleEnterKeySetting	736
SelectTableColumnExtendLeft	709	StyleLinkTo	736
SelectTableColumnExtendRight	709	StyleOff	738
SelectTableRow	710	StyleOn	740
SelectTabularColumn	710	StyleRename	742
SelectToggle	710	StyleRetrieve	742
SelectWord	710	StyleSave	743
SelectWordNext	711	StylesListDlg	744
SelectWordPrevious	711	StyleSystemOn	744
SetDefaultFileFormat	711	StyleType	746
SetDocumentMapping	713	SubdividePage	746
SettingsDlg	713	SubdividePageDlg	747
SetupWritingToolsDlg	714	SubdividePageOff	747
ShiftTabKey	714	SubdocInclude	747

SubstructureExit	748	TableCellUseColumnJust	774
Suppress	748	TableCellUseColumnNumberType	774
SwitchDoc	749	TableCellUseLineColor	775
Tab	749	TableCellVerticalAlignment	775
TabAdd	749	TableColumnAttributeOff	776
TabCenter	750	TableColumnAttributeOn	777
TabDecimal	751	TableColumnDecimalAlignDigits	778
TabDelete	751	TableColumnDecimalAlignDist	778
TabInsert	752	TableColumnFixedWidth	779
TabKey	752	TableColumnJustification	779
TableAdjustBegin	752	TableColumnMarginLeft	780
TableAdjustColumnWidth	753	TableColumnMarginRight	780
TableAdjustEnd	753	TableColumnNumberAlignCurrency	781
TableAdjustPosition	754	TableColumnNumberDateFormat	781
TableAppendRow	754	TableColumnNumberDecimalDigits	782
TableAttributeOff	754	TableColumnNumberFormat	783
TableAttributeOn	755	TableColumnNumberNegNumber	783
TableAutoCalc	756	TableColumnNumberRound	784
TableAutoFill	756	TableColumnNumberSelCurrency	785
TableAutoFormulaEntry	757	TableColumnNumberUseCommas	786
TableBlockOff	757	TableColumnNumberUseCurrency	786
TableBlockOn	757	TableColumnWidth	787
TableBorder	758	TableColumnWidthGrow	788
TableBorderEditBegin	758	TableColumnWidthShrink	788
TableBorderEditEnd	759	TableConvertDlg	788
TableCalculate	759	TableConvertFrom	789
TableCalculateAll	760	TableCopy	789
TableCalculateDlg	760	TableCopyFormula	790
TableCellAttributeOff	760	TableCreate	791
TableCellAttributeOn	761	TableCreateDlg	791
TableCellCopy	762	TableCreateNewDlg	791
TableCellFillColors	762	TableDataFill	792
TableCellFillStyle	763	TableDecimalAlignmentDigits	792
TableCellIgnoreCalculation	764	TableDecimalAlignmentDistance	792
TableCellJustification	765	TableDefaultLine	793
TableCellLine	765	TableDefaultLineColor	794
TableCellLineColor	766	TableDeleteBlock	794
TableCellLock	767	TableDeleteColumn	795
TableCellNumberAlignCurrency	767	TableDeleteDlg	795
TableCellNumberDateFormat	768	TableDeleteRow	796
TableCellNumberDecimalDigits	769	TableDeleteTable	796
TableCellNumberFormat	769	TableDisableCellLocks	797
TableCellNumberNegativeNumber	770	TableEdit	797
TableCellNumberRound	770	TableFill	797
TableCellNumberSelectCurrency	771	TableFormatDlg	798
TableCellNumberUseCommas	772	TableFormatJustification	798
TableCellNumberUseCurrency	773	TableFormatWidths	799
TableCellUseColumnAttribute	773	TableFormula	799

TableFormulaCBar	800	TableUseDefaultLinesColor	822
TabLeft	800	TabRight	822
TableGridLinesToggle	801	TabSet	823
TableGuides	801	TabSetDlg	823
TableHardColumnBreak	801	TemplateCopyObjectDlg	824
TableHardRow	802	TemplateCreate	824
TableHeader	802	TemplateCreateDlg	824
TableInsertColumn	803	TemplateDescriptionDlg	825
TableInsertDlg	803	TemplateDlg	825
TableInsertOneRow	803	TemplateEdit	825
TableInsertRow	804	TemplateSelect	826
TableJoin	804	TextBorderCreate	826
TableJoinCells	805	TextBorderEdit	827
TableLineFillDlg	805	TextBorderEnd	828
TableMarginLeft	805	TextBorderFillStyle	829
TableMarginRight	806	TextBorderStyle	830
TableMove	806	TextBoxCreate	831
TableMoveCopyDlg	807	TextColor	831
TableNameAdd	807	TextShade	832
TableNameChange	808	ThousandsSeparator	832
TableNameCreateDlg	808	ToACreateFullFormDlg	833
TableNameDelete	809	ToADefineDlg	833
TableNameListDlg	809	ToADefinition	833
TableNumberAlignCurrency	809	ToADefinitionCombine	835
TableNumberDateFormat	810	ToADefinitionCreate	836
TableNumberDecimalDigits	810	ToADefinitionDelete	838
TableNumberFormat	811	ToADefinitionMode	839
TableNumberNegativeNumber	812	ToADefinitionRename	839
TableNumberRound	812	ToADefinitionRetrieve	840
TableNumberSelectCurrency	813	ToADefinitionStyle	840
TableNumberTypeDlg	814	ToADefinitionUnderline	842
TableNumberUseCommas	814	ToAEditFullForm	843
TableNumberUseCurrency	815	ToAEditFullFormCBar	843
TablePosition	815	ToAEditFullFormDlg	843
TableQuickSum	816	ToAEditPageFormat	844
TableRetrieve	816	ToAEditSection	844
TableRowHeight	817	ToAEditShortForm	845
TableRowMarginBottom	818	ToAMark	845
TableRowMarginTop	818	ToAMarkShortForm	846
TableRowNumberOfLines	819	ToCDefine	846
TableSelectOff	819	ToCDefineDlg	848
TableSelectOn	819	ToCMark	848
TableSplit	820	Type	849
TableSplitCellsDlg	820	TypeChar	849
TableSplitColumn	820	Typecover	850
TableSplitRow	821	TypeoverKey	850
TableTextToFormula	821	TypesetManualKerningDlg	851
TableUndelete	822	Undelete	851

UndeleteDlg	851	WidowOrphan	859
UnderlineSpaces	852	WindowCascade	859
UnderlineTabs	852	WindowNext	859
Undo	853	WindowPrevious	860
UserFunction	853	WindowTile	860
ViewDraft	853	WordCountDlg	860
ViewOutline	854	WordLetterspaceDlg	860
ViewPage	854	WordLetterSpacing	861
ViewTwoPage	854	WordSpacingJustificationLimits	861
VLineCreate	855	WPCharactersDlg	862
VolumeNumber	855	WPDRAW	862
VolumeNumberDecrement	855	WPGrammatik	862
VolumeNumberDisplay	856	WPSpeller	863
VolumeNumberIncrement	856	WPTThesaurus	863
VolumeNumberMethod	856	WritingToolsDisable	863
WatermarkA	857	ZoomDlg	864
WatermarkB	857	ZoomToFullPage	864
WatermarkDlg	858	ZoomToMarginWidth	864
WatermarkPlacementDlg	858	ZoomToPageWidth	864

System Variables Reference

?Align	867	?BoxCaptionPositionBorder	877
?AutoUpdate	867	?BoxCaptionPositionOffset	877
?BaselineMode	867	?BoxCaptionPositionSide	877
?BetweenTableCodes	867	?BoxCaptionRotation	878
?Binding	868	?BoxChangeLineHeight	878
?BlockActive	868	?BoxContentFilename	878
?BlockProtect	868	?BoxContentHorizontalPosition	879
?BookmarkList	869	?BoxContentInternal	879
?Border	869	?BoxContentPreserveAspectRatio	879
?BorderBottomLine	869	?BoxContentType	880
?BorderColorBlue	870	?BoxContentVerticalPosition	880
?BorderColorGreen	870	?BoxCount	880
?BorderColorRed	870	?BoxCounter	881
?BorderColorShading	871	?BoxEquationColorBlue	881
?BorderCornerRadius	871	?BoxEquationColorGreen	881
?BorderInsideSpacingBottom	871	?BoxEquationColorRed	882
?BorderInsideSpacingTop	872	?BoxEquationColorShading	882
?BorderLeftLine	872	?BoxEquationFont	883
?BorderRightLine	873	?BoxEquationFontSize	883
?BorderSeparatorLine	873	?BoxHeight	883
?BorderSetSpacing	873	?BoxHorizontalAlignment	884
?BorderTopLine	874	?BoxHorizontalAlignTo	884
?BorderUseBorderColor	874	?BoxHorizontalOffset	885
?BoxAttachTo	875	?BoxImageBlackWhiteThreshold	885
?BoxCaptionDefaultInitialStyle	875	?BoxImageBrightness	886
?BoxCaptionDefaultNumberStyle	875	?BoxImageContrast	886
?BoxCaptionFormatting	876	?BoxImageDitherMethod	886
?BoxCaptionPositionAlignment	876	?BoxImageDitherSource	887

?BoxImageFill	887	?CellSmall	901
?BoxImageFlipX	887	?CellSmallCaps	901
?BoxImageFlipY	888	?CellSpecificAttribute	902
?BoxImageHalftoneAngle0	888	?CellSpecificJustification	902
?BoxImageHalftoneAngle1	888	?CellStrikeout	902
?BoxImageHalftoneAngle2	888	?CellSubscript	902
?BoxImageHalftoneAngle3	889	?CellSuperscript	903
?BoxImageHalftoneScreenLPI	889	?CellUnderline	903
?BoxImageInvertColors	889	?CellVeryLarge	903
?BoxImageMonochrome	889	?CenterCurrentPage	903
?BoxImagePageBackground	890	?CenterPages	904
?BoxImageRotation	890	?ChapterNumber	904
?BoxImageScalingX	890	?ChapterNumberMethod	904
?BoxImageScalingY	891	?CharSpaceWidth	904
?BoxImageTranslationX	891	?Column	905
?BoxImageTranslationY	891	?ColumnBorder	905
?BoxLeftColumn	892	?ColumnDefine	905
?BoxNumber	892	?ConfirmCodesDelete	906
?BoxOverlap	892	?ConfirmFormulaDelete	906
?BoxPageOffset	893	?CurrentLabel	906
?BoxRightColumn	893	?CurrentParagraphLevel	906
?BoxStyleName	893	?CurrentPaperName	906
?BoxTextAngle	894	?CurrentPrinter	907
?BoxTextDefaultStyle	894	?CurrentPrinterPort	907
?BoxVerticalAlignment	894	?CurrentPrinterType	907
?BoxVerticalAlignTo	895	?CurrentSubdoc	907
?BoxVerticalOffset	895	?CurrentTemplateName	908
?BoxWidth	895	?CurrentTemplate	908
?BoxWrapping	896	?CurrentWord	908
?BoxWrappingType	896	?DateDay	908
?ButtonBarActive	897	?DateMonth	908
?Cell	897	?DateWeekday	909
?CellAlignment	897	?DateYear	909
?CellBold	897	?DecimalAlignmentCharacter	909
?CellDoubleUnderline	898	?DefaultDocumentExtension	909
?CellExtraLarge	898	?DefaultTemplateExtension	909
?CellFill	898	?DefaultView	909
?CellFine	898	?DisplayMode	910
?CellFormula	899	?DisplayPitch	910
?CellIgnoreWhenCalculating	899	?DocBlank	910
?CellItalics	899	?DocChanged	910
?CellJustification	899	?DocNumber	911
?CellLarge	900	?DocReadOnly	911
?CellLocked	900	?DocumentFont	911
?CellNormal	900	?DocumentInformation	911
?CellOutline	900	?DotLeader	911
?CellRedline	901	?EditScreen	912
?CellShadow	901	?Endnote	912

?EndnoteMinimum	912	?GraphicsLineStyle	925
?EndnoteSpace	912	?GraphicsLineThickness	926
?FeatureBar	912	?GraphicsLineTopSpacing	926
?Fill	913	?GraphicsLineType	927
?FillGradientSteps	913	?GraphicsLineUseColor	927
?FillGradientType	914	?GraphicsLineVerticalAlign	927
?FillType	914	?GraphicsLineVerticalPosition	928
?FillWPG2Pattern	914	?HeaderAOccur	928
?FirstLineIndent	915	?HeaderBOccur	928
?FirstPrintLineNum	915	?HeaderSeparator	929
?Font	915	?HypertextLast	929
?FontBold	915	?HypertextType	929
?FontDoubleUnderline	916	?Hyphenation	929
?FontExtraLarge	916	?Indent	930
?FontFine	916	?IndexDefinition	930
?FontItalics	916	?Initials	930
?FontLarge	917	?InOutline	930
?FontOutline	917	?InterimVersion	931
?FontRedline	917	?Justification	931
?FontShadow	917	?Kerning	931
?FontSize	918	?LabelsToDisplay	931
?FontSmall	918	?Language	932
?FontSmallCaps	918	?LeftChar	932
?FontStrikeout	918	?LeftCode	932
?FontSubscript	918	?LeftHyphZone	932
?FontSuperscript	919	?LeftMarginAdj	933
?FontUnderline	919	?Line	933
?FontVeryLarge	919	?LineHeight	933
?FooterAOccur	920	?LineNumber	933
?FooterBOccur	920	?LineNumbering	933
?FooterSeparator	920	?LineNumberingInterval	934
?Footnote	920	?LineNumberMethod	934
?FootnoteContinued	921	?LineNumCountBlankLines	934
?FootnoteMinimum	921	?LineNumNewspaperCols	935
?FootnotePosition	921	?LineNumPosition	935
?FootnoteRestartsPage	921	?LineSegmentBlue	935
?FootnoteSpace	922	?LineSegmentCount	936
?GraphicsLineBottomSpacing	922	?LineSegmentGreen	936
?GraphicsLineColorBlue	922	?LineSegmentRed	936
?GraphicsLineColorGreen	923	?LineSegmentSetUserData	937
?GraphicsLineColorRed	923	?LineSegmentShading	937
?GraphicsLineColorShading	923	?LineSegmentSpacing	938
?GraphicsLineCount	924	?LineSegmentWidth	938
?GraphicsLineHorizontalAlign	924	?LineSegmentWPG2Pattern	938
?GraphicsLineHorizontalColumn	924	?LineSpacing	939
?GraphicsLineHorizontalPosition	925	?ListDefinition	939
?GraphicsLineLength	925	?MacroExecute	939
?GraphicsLineNum	925	?MacroPlaying	940

?MacroRecord	940	?PowerBarActive	950
?MajorVersion	940	?Random	950
?MarginBottom	940	?RedlineCharacter	950
?MarginLeft	941	?RedlineMethod	951
?MarginRight	941	?RevealCodesActive	951
?MarginTop	941	?RightChar	951
?MergeDone	941	?RightCode	951
?MinorVersion	941	?RightHyphZone	952
?Name	942	?RightMarginAdj	952
?Network	942	?Row	952
?NotFound	942	?RowHeader	952
?Overstrike	942	?RowHeight	952
?Page	942	?RulerBarActive	953
?PageBorder	943	?SaveWorkspace	953
?PageNumberFormat	943	?SecondaryPageNumber	953
?PageNumberingOn	943	?SecondaryPageNumberMethod	953
?PageNumberingValue	943	?SelectedText	954
?PageNumberMethod	944	?SelectedTextSize	954
?PageNumberPosition	944	?ShowBoxes	954
?PageSuppress	944	?SortGetAction	955
?PaperLength	945	?SortGetCase	955
?PaperSizeType	945	?SpaceExpansion	955
?PaperWidth	945	?SpacingBelowPar	955
?ParagraphBorder	945	?StartingLineNumber	956
?ParagraphNumber	946	?StatusBarActive	956
?ParagraphNumberMethod	946	?Substructure	956
?Path	946	?TableDefaultLine	956
?PathBackup	946	?TableGuides	957
?PathCurrent	947	?TableInTable	957
?PathDatabase	947	?TableLastCell	957
?PathDatabaseSupplemental	947	?TabSettings	957
?PathDocument	947	?TemplateFile	958
?PathGraphics	947	?TemplateFileSupplemental	958
?PathGraphicsSupplemental	947	?TextColorBlue	958
?PathHyphenation	948	?TextColorGreen	958
?PathLabels	948	?TextColorRed	958
?PathMacros	948	?TextColumnNumber	959
?PathMacrosSupplemental	948	?TextColumns	959
?PathPrinters	948	?TextShade	959
?PathPrintersSupplemental	948	?ThousandsSeparatorChar	959
?PathProgram	949	?TimeHour	959
?PathShared	949	?TimeHundredth	960
?PathSpreadsheet	949	?TimeMinute	960
?PathSpreadsheetSupplemental	949	?TimeSecond	960
?PathTemplate	949	?ToADefinition	960
?PathTemplateSupplemental	949	?ToAShortFormMarks	960
?PauseKey	950	?ToAShortForms	961
?Position	950	?ToCDefinition	961

?TypeoverActive	961	?UserColorGreen	963
?UndeleteBuffer	961	?UserColorRed	964
?UnderlineSpaces	961	?UserInitials	964
?UnderlineTabs	962	?UserName	964
?UnprintableBottomMargin	962	?VolumeNumber	964
?UnprintableLeftMargin	962	?VolumeNumberMethod	964
?UnprintableRightMargin	962	?WatermarkAOccur	965
?UnprintableTopMargin	963	?WatermarkBOccur	965
?UpdateQuickList	963	?WidowOrphan	965
?UseDefaultDocExt	963	?WritingToolsDisabledState	966
?UserColorBlue	963	?Zoom	966

Coaches Reference

Overview	969	CoachGetRegionInfo	978
CoachSetDialogFilter	970	CoachMessageBox	979
CoachRemoveDialogFilter	972	CoachMessageBoxClose	983
CoachAnimate	972	CoachPrompt	983
CoachFilterAdd	973	CoachPromptClose	985
CoachFilterDestroy	977	Keystring	985
CoachFilterEnable	977	Mouse String	987
CoachFilterDisable	978		

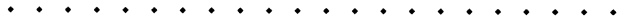
Appendixes

Appendix A: Reserved Words	993
Appendix B: Shipping Macros	995
Appendix C: Windows Only Commands	997
Appendix D: Macro Conversion	1003

Glossary/Index

Glossary	1023
Index	1029

Theory and Practice



Introduction

WordPerfect's command-based macro language consists of statements that, when compiled, perform tasks. Statements are instructions to the computer. You combine statements in macros to automate routine tasks and simplify large ones. See *Compiler* and *Command-Based Macro Language* in *Macro Facility*.

What's in this Manual?

This manual discusses the main statement types used to write a macro: assignments, conditions, loops, comments, and commands. See *Macro Language Concepts* for brief descriptions, and *Variables, Constants, Operators, and Expressions*, and *Macro Control Statements* for detailed descriptions and examples.

This manual also presents five full-length macro examples with line-by-line comments. Three of the examples, *ASSERT.WCM*, *CALLBACK.WCM*, and *CB_SCBAR.WCM*, have no purpose except to show some advanced macro techniques. Two other examples, *VIEWBARS.WCM* and *CALENDAR.WCM*, are practical examples that demonstrate additional useful techniques. To edit or play the macros, copy them from the WordPerfect for Windows 6.0a Online Macros Manual.

User's Guide

This section includes topics that describe a macro and how to create one. If you are new to macro writing, *Writing Macros* suggests a macro writing style to help you get started. An experienced macro writer may want to read *Getting Started* (below), then skip to the reference sections.

References

The Programming and Product Commands, System Variables, and Coaches Reference sections describe each macro command and system variable. Topics include purpose, syntax, return value, comments, parameters, examples, and references to related commands.

Appendixes

- Appendix A: Reserved Words
- Appendix B: Shipping Macros
- Appendix C: Windows Only Commands
- Appendix D: Macro Conversions

Getting Started

In this manual,

- ♦ Italicized cross-references, such as See *Subroutines* in *Macro Control Statements*, refer to topic and section headings. See *Writing Macros* refers to a section heading.
- ♦ See *CHAIN* in *Programming Commands Reference* refers to a programming command. See *AboutDlg* in *Product Command Reference* refers to a product command. See *?MarginLeft* in *System Variables Reference* refers to a system variable. Programming commands, product commands, and system variables appear in alphabetical order in their respective sections and in the index.
- ♦ Programming commands, such as *SWITCH*, are all uppercase to distinguish them from product commands. You can enter commands in uppercase, lowercase, or mixed case. The compiler is not case sensitive (see *Compiler* in *Macro Facility*).

- Product commands, such as AboutDlg or MarginLeft, are mixed case.
- System variables, such as ?DocBlank or ?PaperWidth, are mixed case and usually begin with a question mark.
- Italicized words in a syntax definition are user-defined values.

Macro Hints:

- Wrapping does not affect macro execution.
- Do not insert spaces in a command name unless the space is part of the syntax.
- Do not use hard returns, tabs, or indents within a character expression (a character string enclosed in double quotation marks). See *Character Expressions* in *Variables, Constants, Operators, and Expressions*.
- You can use spaces, tabs, indents, and hard returns before and after commands, or between parameters, to make a macro easier to read.
- Smart Quotes in a macro create a compile-time syntax error. To disable, choose Tools, QuickCorrect, then deselect Enable SmartQuotes.
- Command syntax must be correct for a macro to compile. Common errors include missing or incorrectly used colons (:), semicolons (;), and double quotation marks (").
- A macro is automatically compiled the first time it plays. You can also compile macros from Macro Facility (see *Compiler* in *Macro Facility*), or choose Save & Compile on the Macro feature bar (see *Macro Feature Bar* in *Writing Macros*).
- Begin macros with an APPLICATION statement to identify, for the compiler, the application used in the macro. You can include more than one APPLICATION statement (see *APPLICATION* in *Programming Commands Reference*).

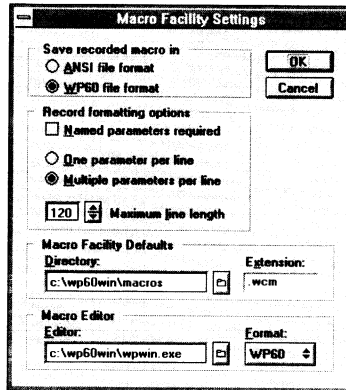
Macro Facility

Macro Facility is an application that ships with WordPerfect for Windows 6.0, that records, plays, compiles, and converts macros created for a Shared Code application (Shared Code refers to programming code used by WordPerfect Windows applications).



Macro Facility automatically records, plays, compiles, and converts macros when you choose one of these options from a Shared Code application. If you use Macro Facility to play a macro for an application that is not running, Macro Facility automatically starts the application.

Set Macro Facility options, including the default macro directory, editor, and extension in the Macro Facility Settings dialog box. To display the dialog box, choose Settings, then Options:



Compiler

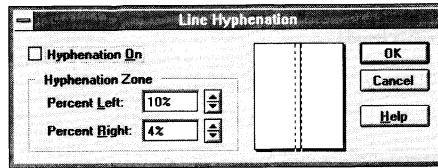
A computer's central processing unit (CPU) executes instructions written in machine (computer) language. Machine language consists of *binary digits* (bits) 0 and 1. For example, the first three letters of the alphabet in binary notation are 1000001, 1000010, and 1000011. The binary result of $4 + 5$ is 1001. Since binary digits are hard to work with, English-based programming languages such as Basic, Pascal, and C were created. Programs are written with an editor or word-processor and saved as *source files*. *Compilers* create *object files*, which are copies of source files translated into machine language. Computers can only execute object files.

Macro languages are similar to programming languages. Macros are written with an editor or word-processor and saved as a source file. When Macro Facility compiles a source file, however, it does not create a separate object file. Instead, it saves the *object* in the source file header, where you cannot see it. When you edit a source file, you must recompile it to update the object. WordPerfect can only execute (play) a source file with an up-to-date object. A compiled source file, or macro, contains instructions Macro Facility interprets and sends to the appropriate application for execution when you play the macro (see *Playing Macros*).

Command Based Macro Language

A command-based macro language records keystrokes and/or mouse actions as commands. It does not record the actual keystrokes. For example, to display the Line Hyphenation dialog box, choose Layout, Line, then Hyphenation. If you record these

steps, your macro contains a product command named `LineHyphenationDlg`, which displays the following dialog box:



See *Product Commands* in *Macro Language Concepts*, and *Product Commands Reference*.

Compiler Errors

The compiler stops and displays a message when it discovers a macro syntax error (see *Syntax* in *Macro Language Concepts*). The message contains information about the error and its location (the compiler makes a best guess and may not always be accurate). Choose `Cancel Compilation` to dismiss the message box, or `Continue Compilation` to test for other errors. You must correct all errors before a macro compiles. See *Macro Errors* in *Writing Macros* for common syntax errors.

Run-time Errors

Run-time is the same as execution time. Run-time errors occur while the macro is executing. Referencing a variable that has not been assigned a value causes a run-time error. See *Macro Errors* in *Recording Macros*.

Macro Language Concepts

This section introduces the main statement types used to write a macro. Begin here if you are new to macro writing, or skip to the command reference sections if you are an experienced macro writer. The topics below are presented in detail in *Variables*, *Constants*, *Operators*, and *Expressions*, and in *Macro Control Statements*.

Macros

Automate WordPerfect tasks, such as setting margins, selecting a font, or creating a merge file. Macro tasks consist of instructions called statements. Statements can be *assignments*, *conditions*, *loops*, *comments*, or *commands*. For example,

```
MarginLeft (1.0")  
MarginRight (1.0")
```

are commands that set one-inch left and right margins (the task). Each command represents one instruction or statement.

The simplest macro consists of only one statement. Complex macros have hundreds of statements. The sequence of statements determines how a macro performs its tasks.

Syntax

Refers to rules that govern the form of macro statements. For example,

```
Type("John Doe")
```

types John Doe. The next example,

```
Type("John Doe"
```

lacks a closing parenthesis. The syntax is not correct and produces an error message. See *Programming Commands Reference*, *Product Commands Reference*, and *System Variables Reference* for the correct syntax of macro commands and system variables. For common syntax errors, see *Macro Errors* in *Writing Macros*.

Assignment Statements

Assign the value of an expression to a variable.

```
x := "John Doe"
```

Result: x equals John Doe

```
y := 5
```

Result: y equals 5

```
z := 3 + 4
```

Result: z equals the result of 3 + 4

The assignment operator (`:=` or `=`) assigns the value of a right operand expression to a left operand variable. You can also assign values to variables using the `ASSIGN` programming command (see `ASSIGN` in *Programming Commands Reference*).

Expressions

Represent values, and can be arithmetic, numeric, measurement, relational, logical, bitwise, or a character string. See *Expressions* in *Variables, Constants, Operators, and Expressions*.

Variables

Represent data that can change during macro execution. Data may include any type of expression, but only one expression at a time. Variable names are user-defined, are not case sensitive, must begin with a letter, can include any other combination of letters or numbers, and are limited to 50 characters. For example, if you assign 1.0" to variable `vMargin`, the following commands set one-inch top and bottom margins:

```
MarginTop (vMargin)  
MarginBottom (vMargin)
```

The lowercase `v` in `vMargin` is an optional convention to identify variables. For more information, see *Variables* in *Variables, Constants, Operators, and Expressions*.

System Variables

Return current system information such as font names, margins, and default directories. System variables are defined by WordPerfect and most begin with a question mark. For example,

```
vMacroPath := ?PathMacros
```

assigns the path and name of the default macros directory to variable `vMacroPath`. If you change the directory, system variable `?PathMacro` returns an updated value. See *System Variables Reference*.

Conditional Statements Execute a statement or group of statements (statement block) when a specified condition is met. If a macro displays a list of options, it executes a statement block depending on which option the user chooses. Conditional statements include CASE, IF, IFPLATFORM, and SWITCH. See *Conditional Statements* in *Macro Control Statements*.

Loop Statements Execute a statement or statement block a specified number of times, until an expression is true, or while an expression is true. The macro then exits the loop and continues to the next statement. Loop statements include FOR, FORNEXT, FOREACH, REPEAT, and WHILE. See *Loop Statements* in *Macro Control Statements*.

Comment Statements Contain notes and other information that do not affect macro execution. Use comment statements to explain the purpose of your macro, describe its components, or to prevent a statement from executing. Comment statements help if you have to modify a macro months after it is written, or if someone else has to understand your macro. A comment begins with // and ends with a hard return [HRt]. See // (Comment) in *Programming Commands Reference*.

Command Statements Consist of a name, and optionally one or more parameters. Commands represent an instruction to the parent application. See *Product Commands Reference* and *Programming Commands Reference*.

Command Name

Command names often describe a feature, such as Font, MarginLeft, Advance, and FootnoteOptions. Command names are not case sensitive and usually do not contain spaces. Exceptions include programming commands that call a subroutine, such as CASE CALL or ONCANCEL CALL (see *Subroutines* below).

A macro can use more than one application. Commands to the non-default application require a *product prefix*, which is specified in an APPLICATION statement. In this example,

```
A1.AboutDlg ()
```

A1 (followed by a period) is the product prefix. It identifies the application for the compiler to use. See *Compiler* in *Macro Facility*, and *APPLICATION* in *Programming Commands Reference*.

Parameters

Contain values that are passed to product commands and user-defined subroutines. In this example,

```
Advance (Where: AdvanceDown!; Amount: 1.0")
```

Advance is the command name, and AdvanceDown! and 1.0" are parameter data. This command advances the insertion point down one inch. Parameter names, such as Where and Amount, followed by a colon, are optional.

A parameter set begins with an open parenthesis and ends with a closed parenthesis. Multiple parameters are separated with a semicolon. Some commands have no

parameters, and others have many (see *AboutDlg* and *PaperSizeSelect* in *Product Commands Reference*). Parentheses are optional for commands with no parameters. This is not true of user-defined functions and procedures (see *Subroutines* below and in *Macro Control Statements*).

Repeating parameters begin with an open brace {}, end with a closed brace {}, and are separated by a semicolon. For example, MENU may list up to twenty-six items. The following example lists only three:

```
MENU(vChoice; Digit; 10; 10; {"Item 1"; "Item 2"; "Item 3})
```

Product Commands

Execute product features. For example,

```
FontSize (12.0p)
```

sets the current font size to 12-point. The Margin and Advance commands in the examples above are product commands. See *Product Commands Reference*.

Programming Commands

Direct the execution of a macro. For example,

```
IF(x = "A")
  ButtonBarCreateDlg
ELSE
  ButtonBarOptionsDlg
ENDIF
```

displays the Create Button Bar dialog box if x equals the value A, or the Button Bar Options dialog box if x has any other value. IF, ELSE, and ENDIF are programming commands. ButtonBarCreateDlg and ButtonBarOptionsDlg are product commands.

In this manual, programming commands appear in uppercase, product commands in mixed case. However, product and programming commands are not case sensitive.

Subroutines

Consist of a statement or group of statements (statement block) that are executed when the subroutine is called (see *Calling Statements* in *Macro Control Statements*). Subroutines include functions, procedures, and labels.

```
CALL(SubExample)
```

```
...(other statements)...
```

```
LABEL(SubExample)
```

```
...statement block...
```

```
RETURN
```

Explanation: The calling statement CALL(SubExample) calls (directs macro execution to) the subroutine LABEL(SubExample). RETURN directs macro execution to the statement that follows CALL(SubExample). Statements in a subroutine are accessible to any part of a macro, and can be called any number of times during execution. See *Subroutines* in *Macro Control Statements*.

Variables, Constants, Operators, and Expressions

Variables and constants are joined by operators to form expressions. Expressions often compute values that determine how control statements direct macro execution (see *Macro Control Statements*).

Variables

Point to a memory cell where data is stored. The variable name represents the data's address or location. An address can store only one data item at a time, which can be of any type and can change during macro execution. Use `DECLARE`, `LOCAL`, `GLOBAL`, or `PERSIST` programming commands to declare variables.

Adjacent memory cells, associated by a name, are called an array. Use `DECLARE`, `LOCAL`, `GLOBAL`, or `PERSIST` programming commands to declare an array name and the number of cells, or elements, associated with it. Assign a value to an array element as you assign a value to a variable. For example,

```
DECLARE Test[10]
ASSIGN(Test[1]; 5)
```

creates a 10-element array named `Test`, and assigns 5 to element 1. Array elements can contain different data types, such as numbers or character strings. See *DECLARE* in *Programming Commands Reference* for array examples.

IMPORTANT: A run-time error occurs if you reference a variable or array element that has not been assigned a value. See *VARERRCHK* in *Programming Commands Reference*.

Variable and array names are user-defined, are not case sensitive, must begin with a letter, can include any other combination of letters or numbers, and are limited to 50 characters. The following examples assign a right operand expression to a left operand variable (a variable can also be an array element):

```
x := "John Doe"
```

Result: x equals John Doe (character string)

```
vLeftMargin := 5i
```

Result: vLeftMargin equals 5I (measurement expression)

```
ResultOfOperation := 3 + 4
```

Result: ResultOfOperation equals 7 (numeric expression)

```
z := z + 1
```

Result: z equals the value of z + 1 (Since a variable can contain only one value at a time, the original value of z is lost unless previously assigned to another variable.)

```
x := y > 1
```

Result: x equals the result of relational expression `y > 1` (x equals `True` if y contains a value greater than 1, or `False` if y contains a value less than or equal to 1. `True` and `False` are Boolean data that represent numeric constants 1 and 0.)

The following example evaluates the result of $y > 1$, without assigning the result to a variable. The computer beeps if the value of y is greater than 1 (if the result of expression $y > 1$ equals True). The first statement after ENDIF is executed if the value of y is less than or equal to 1 (if the expression result equals False).

```
IF(y > 1)
    BEEP
ENDIF
```

INDIRECT

Creates a variable name out of a combination of character strings and/or numbers (see *INDIRECT* in *Programming Commands Reference*). You can use INDIRECT wherever you can use a variable.

INDIRECT can also create a LABEL name out of a combination of character strings and/or numbers. See *Subroutines* in *Macro Control Statements*.

DECLARE

Creates local variables and arrays (only pertain to the current macro). An array can have up to 10 dimensions, and up to 32,767 elements per dimension depending on the amount of available memory (see DECLARE in *Programming Commands Reference*).

LOCAL

Creates local variables and arrays (only pertain to the current macro). Variables declared in user-defined functions and procedures are local to those subroutines. When you create a variable by assigning it a value, the variable is automatically declared local (see *LOCAL* in *Programming Commands Reference*).

GLOBAL

Creates variables and arrays for the current macro and macros called by RUN or CHAIN (see *CHAIN*, *GLOBAL*, and *RUN* in *Programming Commands Reference*).

PERSIST

Creates variables and arrays for any Shared Code macro as long as Shared Code is running (see *PERSIST* and *PERSISTALL* in *Programming Commands Reference*).

MacroDialogResult

An implicit variable (one defined by the application) that contains the *Control* value of the button that dismisses a dialog box. See *Creating a Dialog Box* in *Dialog Boxes*, and *DIALOGDEFINE* in *Programming Commands Reference*.

ErrorNumber

A special variable that contains the error value of a Cancel, Error, or Not Found condition. See *ASSERT* in *Programming Commands Reference*, and *ASSERT.WCM* example (line 44) in *Macro Menus*.

Constants

Represent data that cannot change during macro execution. For example,

```
SWITCH(x)
    CASEOF 1: CALL(Start)
    CASEOF 2: CALL(Stop)
ENDSWITCH
```

contains two CASEOF statements, followed by numeric constants 1 and 2. If variable x equals 1, a subroutine named Start is called. If x equals 2, a subroutine named Stop is called. See *Conditional Statements and Subroutines* in *Macro Control Statements*, and *SWITCH* in *Programming Commands Reference*.

Unary Operators

A symbol or word that represents an operation on only one operand or expression.

Operator Action

+	Multiplies an operand by +1
-	Multiplies an operand by -1
NOT	Inverts the result of relational and logical expressions

Unary plus/minus (+/-) examples

+5

Result: +5 is the result of 5 * +1

-10

Result: -10 is the result of 10 * -1

NOT example

```
x := 5
y := (x < 10)
z := NOT(x < 10)
```

Explanation: True is assigned to y (x is less than 10). False is assigned to z (inverted result of expression x < 10).

NOT example in an IF statement

```
x := 5
z := (x < 10)
IF(NOT(z))
  BEEP
ELSE
  QUIT
ENDIF
```

Explanation: The result of expression x < 10 is assigned to variable z. BEEP causes the computer to beep if the inverted value of z equals True. Since the inverted value of z equals False, there is no beep, and the ELSE statement is executed (QUIT ends the macro).

NOT example in shorthand notation

```
x := 5
IF(NOT(x < 10))
  BEEP
ELSE
  QUIT
ENDIF
```

Explanation: IF beeps the computer if the inverted result of expression $x < 10$ equals True. Since the result of $\text{NOT}(x < 10)$ equals False, there is no beep, and the ELSE statement is executed (QUIT ends the macro).

Previous example without NOT operator

```
x := 5
IF(x > 10)
  BEEP
ELSE
  QUIT
ENDIF
```

Explanation: If the result of expression $x > 10$ equals True, beep. Since the result equals False (the value of x is less than 10), the ELSE statement is executed (QUIT ends the macro).

Binary Operators

A symbol that represents an operation on two operands or expressions. For example,

```
x := 3 + 4
```

contains a plus operator (+) which adds the operands 3 and 4. The assignment operator (:=) assigns the result of the arithmetic expression $3 + 4$ to variable x.

Arithmetic Operators

A symbol or word that represents a mathematical operation on two operands.

Operator	Action
*	Multiplication
/	Division
-	Substraction
+	Addition
%	Floating point modulus division (returns floating point division remainder)
MOD	Integer modulus division (returns integer division remainder)
DIV	Integer division (returns integer portion of integer division)

Floating point % examples

```
x := 10.1 % 3
```

Result: x equals 1.1

```
x := 9 % 3
```

Result: x equals 0

MOD examples

```
x := 10 MOD 3
```

Result: x equals 1

x := 10.1 MOD 3

Result: error (cannot use MOD on real numbers)

DIV examples

x := 10 DIV 3

Result: x equals 3

x := 9 DIV 3

Result: x = 3

x := 9.1 DIV 3.5

Result: error (cannot use DIV on real numbers)

Relational Operators

A symbol that represents a relational operation on two operands. The operation result equals True or False.

Operator	Action
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
=	Equal to
<>	Not equal to

Greater than (>) and less than (<) examples

x := 10
z := (x > 5)

Result: z equals True (x is greater than 5)

z := (x < 5)

Result: z equals False (x is not less than 5)

Not equal (<>) example

x := 10
z := (x <> 5)

Result: z equals True (x is not equal to 5)

Greater than or equal to (>=) example

x := 10
IF(x >= 10)
 BEEP
ELSE
 QUIT
ENDIF

Explanation: The computer beeps because the result of expression $x \geq 10$ equals True (x equals 10). The ELSE statement is skipped (QUIT does not end macro).

Less than or equal to (<=) example

```
x := 20
IF(x <= 10)
    BEEP
ELSE
    QUIT
ENDIF
```

Explanation: The computer does not beep because the result of expression $x \leq 10$ equals False (x is greater than 10). The ELSE statement is executed (QUIT ends macro).

Logical Operators

Words that represent a logical relationship between conditions, or that invert a condition. The condition is the result of a relational expression. The operators are listed in order of precedence.

Operator	Action
----------	--------

NOT	NOT is a unary operator that inverts the result of a relational expression.
-----	---

AND	Combines two relational expressions. Each expression must be true for the logical expression to be true.
-----	--

XOR	Combines two relational expressions. Only one expression can be true for the logical expression to be true. If both are true or both are false, the logical expression is false. The XOR operator is also named exclusive OR.
-----	---

OR	Combines two relational expressions. Only one expression need be true for the logical expression to be true. The OR operator is also named Inclusive OR.
----	--

NOT example

```
x := 8
IF((x < 10) AND NOT (x = 5))
    BEEP
ENDIF
```

Explanation: The result of the logical expression equals True (x is less than 10 AND NOT x is equal to 5, or x is not equal to 5).

AND example

```
x := 1
y := 2
z := ((x = 1) AND (y = 2))
```

Result: z equals True (the logical AND expression is true because the relational expressions $x = 1$ and $y = 2$ are true)

XOR example

```
x := 1
y := 2
z := ((x = 0) XOR (y = 2))
```

Result: z equals True (only one relational expression is true)

```
z := ((x = 1) XOR (y = 2))
```

Result: z equals False (both relational expressions are true)

```
z := ((x = 0) XOR (y = 1))
```

Result: z equals False (both relational expressions are false)

XOR example in shorthand notation

```
x := 1
y := 2
IF((x = 0) XOR (y = 2))
    BEEP
ENDIF
```

Explanation: The result of the logical XOR expression equals True because relational expression $x = 0$ is false and relational expression $y = 2$ is true.

OR example

```
x := 1
y := 2
z := ((x = 1) OR (y = 5))
```

Result: z equals True (the logical OR expression is true because relational expression $x = 1$ is true)

Bitwise Operators

A symbol that represents a bitwise operation on two integer operands. The operators are listed in order of precedence.

Operator	Action
<code>~</code>	Bitwise unary NOT (one's complement) toggles a binary value (1 to 0, and 0 to 1).
<code>&</code>	Bitwise AND results in 1 if both operand bits are 1, and 0 if not.
<code> </code>	Bitwise Inclusive OR results in 1 if both operands are 1 or operands do not match. Results in 0 if both operands are 0.
<code>^</code>	Bitwise Exclusive OR (XOR) results in 0 if operands match, and 1 if not.
<code><<</code>	Shift bits left.
<code>>></code>	Shift bits right.

Unary NOT (~) example

DEC	BIN
~-15	1111111111110001
----	-----
14	0000000000011110

Explanation: Bitwise NOT (one's complement) toggles bits from 1 to 0, and 0 to 1.

AND (&) example

DEC	BIN
1000	1111101000
&31	0000011111
----	-----
8	0000001000

Explanation: Bitwise AND turns bits off (result is 0) if one operand is 0. Bits are left on (result is 1) if both operands are 1.

Inclusive OR (|) example

DEC	BIN
1000	1111101000
27	0000011011
----	-----
1019	1111111011

Explanation: Bitwise OR turns bits on (result is 1) if one operand is 1. Bits remain off (result is 0) if both operands are 0.

Exclusive OR (^) example

DEC	BIN
1000	1111101000
^40	0000101000
----	-----
960	1111000000

Explanation: Bitwise XOR turns on bits (result is 1) if the operands are different. It turns off bits (result is 0) if the operands are the same.

Shift left (<<) example

DEC	BIN
500	0111110100
<<1	0000000001
----	-----
1000	1111101000

Explanation: Shift bits one position left, and insert one 0 bit at the right end of the binary numeral. Multiplies the original value by 2. If the shifted value is so large the top bit is set, 1 shift left shifts the top bit off.

Shift right (>>) example

```
1000  1111101000
>>1   0000000001
-----
500   0111110100
```

Explanation: Shift bits one position right, and insert one 0 bit at the left end of the binary numeral. Divides the original value by 2.

Operator Precedence

The following table shows operator precedence, which is applied to expressions with two or more operators.

Order	Operators
1	Parentheses (), unary minus (-), unary plus (+), bitwise NOT (~), logical NOT
2	Multiply (*), divide (/), modulus division (% or MOD), integer division (DIV)
3	Add (+), subtract (-)
4	Shift left (<<), shift right (>>)
5	Less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), not equal (<>), equal (=)
6	Bitwise AND (&), bitwise OR (), bitwise XOR (^)
7	Logical AND, logical XOR
8	Logical OR

The following rules apply to operators:

- ♦ Operators with the same precedence are evaluated from left to right.
- ♦ Operators inside parentheses are evaluated before operators outside parentheses.
- ♦ Operators inside nested parentheses are evaluated from the innermost parentheses out.

Operator Precedence Examples

```
x := ((50 * 5 + 50) * 3 + 100)
```

Result: x equals 1000

Explanation: $50 * 5 = 250$, $250 + 50 = 300$, $300 * 3 = 900$, $900 + 100 = 1000$.

```
x := ((50 * (5 + 50)) * 3 + 100)
```

Result: x equals 8350

Explanation: $5 + 50 = 55$, $55 * 50 = 2750$, $2750 * 3 = 8250$, $8250 + 100 = 8350$.

```
x := ((50 * 5 + 50) * (3 + 100))
```

Result: x equals 30900

Explanation: $50 * 5 = 250$, $250 + 50 = 300$, $300 * 103 = 30900$.

Expressions

Macros support numeric, measurement, character, arithmetic, relational, logical, and bitwise expressions. Most commands and function calls can be used in an expression if they return a value. See *USE* example in *Programming Commands Reference*.

Numeric Expressions

Numeric constants or variables, or a combination of the two joined by a numeric operator. A numeric operator is a word or symbol that operates on numeric data (see *Unary* and *Binary Operators* above).

Given that x equals 3, the following examples are valid numeric expressions:

x

Explanation: Variable containing a numeric value.

5

Explanation: Numeric constant.

x * 5

Explanation: Expression (multiply x and 5). The expression x * 5 is also an arithmetic expression.

+5

Explanation: Unary plus constant.

-(x + 10)

Explanation: Unary minus expression (negate the result of x plus 10)

Measurement Expressions

Numeric constants or variables containing a measurement value, or combination of the two joined by a numeric operator.

Given that z equals 4i, the following examples are valid measurement expressions:

z

Explanation: Variable that contains a measurement value of 4 inches.

z * 10i

Explanation: Expression (multiply z and 10 inches).

-z

Explanation: Unary minus (negative 4 inches).

5c

Explanation: Constant (5 centimeters).

The valid units of measure are:

Unit of Measure	Measurement Type
"	inches
i	inches

Unit of Measure	Measurement Type
c	centimeters
m	millimeters
p	points (72 per inch)
w	WP unit (1200 per inch)

Radix Base Conversions

The radix is the base of a number system. Base conversion options are:

Identifiers	Description
h or x	Hexadecimal (radix is 16)
o	Octal (radix is 8)
b	Binary (radix is 2)

Radix Examples

```
x := 1Ah
```

Result: x equals 26 (h is the Hex identifier)

```
x := 1111b
```

Result: x equals 15 (b is the Binary identifier)

```
x := 44o
```

Result: x equals 36 (o is the Oct identifier)

All radix bases must begin with a number. Precede non-numeric hex digits with 0.

```
x := 0Ah
```

Result: x equals 10

Character Expressions

A character string (letters, numbers, or keyboard symbols) or variable, or combination of the two concatenated by the plus operator (+) or compared by a relational operator such as > (greater than). Character strings and character strings assigned to variables must be enclosed in double quotation marks.

Enclose a character constant in single quotation marks to specify a numeric value (see *CTON* in *Programming Commands Reference*). For example,

```
x := 'A'
```

Result: x equals 65

```
x := 'A' + 'B'
```

Result: x equals 131 (65 + 66)

A character string with double quotation marks is enclosed with two sets of double quotation marks. For example,

```
x := "His name is ""John"" Doe"
```

Result: x equals His name is "John" Doe

```
x := ""John Doe""
```

Result: x equals "John Doe" (outside double quotation marks are always required)

The following examples are valid character expressions (notice the space after Joe and before Jr.):

```
"John Doe"
```

Explanation: John Doe is a character string.

```
z := "Joe " + "Doe"
```

Explanation: Two character strings concatenated and assigned to variable z (z equals John Doe).

```
x := z + ", Jr."
```

Explanation: Variable z and character string concatenated, and assigned to variable x (x equals John Doe, Jr.).

If you concatenate a character string and a number, the number is converted to a character string. If you concatenate a numeric character and a number, the numeric character is converted to a number and the two are added.

```
x := "A" + 1
```

Result: x equals A1 (character string)

```
x := "1" + 1
```

Result: x equals 2 (number)

```
x := ("A" + (1 + 3))
```

Result: x equals A4 (the result of a mathematical operation ($1 + 3 = 4$) converted to a character string before being concatenated to A)

Compare the previous example with the following examples (operation precedence is left to right):

```
x := ("A" + 1 + 3)
```

Result: x equals A13 (character string— numbers converted and not added)

```
x := (1 + 3 + "A")
```

Result: x equals 4A (character string— numbers added and then converted to a character string)

Arithmetic Expressions

A statement that represents an arithmetic operation, or one that contains two operands joined by an arithmetic operator. The operation result is a numeric value.

```
x := 1 + 2
```

Result: x equals 3

```
x := 3 * 3
```

Result: x equals 9

```
x := "2" * 3 * 4
```

Result: x equals 24 (2 converted to a number then multiplied)

```
x := "A" * 2
```

Result: error (cannot multiply letters and numbers)

Relational Expressions

A statement that represents a relational operation, or one that contains two operands joined by a relational operator. The operation result equals True or False.

Given that x equals 5,

```
z := (x > 6)
```

Result: z equals False (5 is less than 6)

```
z := (x >= 5)
```

Result: z equals True (5 equals 5)

```
z := ("Ab" > "Bb")
```

Result: z equals False (Ab is less than, or comes before, Bb)

```
z := ("Ab" <> "Bb")
```

Result: z equals True (Ab is not equal to Bb)

Given that x equals "A", y equals "B", and z equals "a", the result of the following expressions (True or False) is assigned to variable w:

```
w := (x > y)
```

Result: w equals False (uppercase A is less than, or comes before, uppercase B)

```
w := (x > z)
```

Result: w equals True (uppercase A is greater than, or comes after, lowercase a)

See *Relational Operators*, and examples of shorthand notation in *Unary* and *Binary Operators* (above).

Logical Expressions

A statement that represents a logical operation, or one that contains two relational expressions joined by a logical operator. The operation result equals True or False.

Given that x equals 10, y equals 5, and z equals 20, the result of the following expressions (True or False) is assigned to variable w:

```
w := ((x > y) AND (y < z))
```

Result: w equals True (both relational expressions are true)

```
w := ((x < y) AND (y < z))
```

Result: w equals False (first relational expression is false)

w := NOT(y > z)

Result: w equals True (5 is not greater than 20)

w := ((x > 5) AND (y < 20) AND (z = 20))

Result: w equals True (all relational expressions are true)

w := (((x < 5) AND (y > 20)) OR (z = 20))

Result: w equals True (z = 20 is true)

w := (((x < 5) AND (y > 20)) OR NOT (z = 20))

Result: w equals False (all relational expressions are false)

See *Logical Operators* and examples of shorthand notation in *Unary* and *Binary Operators* (above).

Bitwise Expressions

A statement that represents a bitwise operation, or one that contains two operands joined by a bitwise operator.

Bitwise NOT (~) example

x := ~(-15)

Result: x equals 14 (one's complement)

x := ~(-15) + 1

Result: x equals 15 (two's complement)

Bitwise AND (&) example

x := 65535 & 535

Result: x equals 535

Previous example in binary notation:

```

1111111111111111 (binary 65535)
& 0000001000010111 (AND binary 535)
-----
0000001000010111 (result is binary 535)

```

Explanation: Bitwise AND operation result is 1 if both operands are 1, or 0 if not.

Bitwise inclusive OR (|) example

x := 65535 | 535

Result: x equals 65,535

Previous example in binary notation:

```

1111111111111111 (binary 65535)
| 0000001000010111 (OR binary 535)
-----
1111111111111111 (result is binary 65535)

```

Explanation: Bitwise OR operation result is 1 if one operand is 1, or 0 if both operands are 0.

Bitwise XOR (^) example

```
x := 65535 ^ 535
```

Result: x equals 65,000

Previous example in binary notation:

```
1111111111111111 (binary 65535)
^ 0000001000010111 (XOR binary 535)
-----
1111110111101000 (result is binary 65,000)
```

Explanation: Bitwise XOR operation result is 0 if both operands are 0 or 1, or 1 if operands do not match.

Bitwise shift left (<<) example

```
x := 65535 << 1
```

Result: x equals 131,070

Previous example in binary notation:

```
1111111111111111 (binary 65535)
-----
<< 1111111111111110 (result is binary 131,070)
```

Explanation: Bitwise shift left operator shifts bits one position left, and inserts one 0 bit at the right end of the binary numeral. The result is the original value multiplied by 2.

Bitwise shift right (>>) example

```
x := 65535 >> 1
```

Result: x equals 32,767

Previous example in binary notation:

```
1111111111111111 (binary 65535)
-----
>> 0111111111111111 (result is binary 32,767)
```

Explanation: Bitwise shift right operator shifts bits one position right, and inserts one 0 bit at the left end of the binary numeral. The result is the original value divided by 2.

Macro Control Statements

Control statements alter the sequential execution of macro commands. For example,

```
MarginLeft(1.0")
MarginLeft(2.0")
```

sets a one-inch left margin and then a two-inch left margin. The second command overrides the first. A control statement can select which `MarginLeft` to execute depending on a condition being true.

```
SWITCH(Test)
  CASEOF 1: MarginLeft(1.0")
  CASEOF 2: MarginLeft(2.0")
  DEFAULT: MarginLeft(3.0")
ENDSWITCH
```

Explanation: If variable *Test* equals 1, set a one-inch left margin. If *Test* equals 2, set a two-inch left margin. If *Test* equals any value except 1 or 2, set a three-inch left margin. The value of *Test* can be determined by a programming command such as `MENU` or `GETNUMBER`. Macros support conditional control statements, loop control statements, and miscellaneous control statements such as `CALL`, `GO`, and `QUIT`. See command names in *Programming Commands Reference*.

Conditional Statements

Execute a statement block when an expression is true, or a variable matches a constant. Macros support the following conditional statements:

CASE

Executes a LABEL statement when *Test* (user-defined variable) matches a constant value.

```
CASE(Test; {1; Start; 2; Next}; Last)
```

```
...(other statements)...
```

```
LABEL(Start)
...statement block...
```

```
LABEL(Next)
...statement block...
```

```
LABEL(Last)
...statement block...
```

Explanation: If *Test* matches 1, call LABEL(Start). If *Test* matches 2, call LABEL(Next). If there is no match, call LABEL(Last). CASE CALL is a similar statement which expects a RETURN after LABEL. See CASE and CASE CALL in *Programming Commands Reference*.

IF-ELSE-ENDIF

Executes a statement block when an expression is true.

```
IF(x = 5)
...statement block...
```

```
ELSE
...statement block...
ENDIF
```

Explanation: Execute the first statement block if the result of expression $x = 5$ equals True (if x equals 5). If the result equals False, execute the second statement block. ELSE is optional.

```
IF(Expression)
...statement block...
ENDIF
```

Explanation: If *Expression* is true, execute statement block. If not true, execute the first statement after ENDIF. *Expression* must evaluate to True or False. See *Relational Expressions in Variables, Constants, Operators, and Expressions* and *IF in Programming Commands Reference*.

IFPLATFORM-ENDIFPLATFORM

Compiles a statement block when the current platform matches a platform specified in the IFPLATFORM argument.

```
IFPLATFORM(DOS; WIN)
...statement block...
ENDIFPLATFORM
```

Explanation: If the current platform is either DOS or Windows, compile the statements between IFPLATFORM and ENDIFPLATFORM. Otherwise, ignore the statements.

SWITCH-ENDSWITCH

Executes a statement block when *Test* matches *Selector*.

```
SWITCH(Test)
CASEOF Selector1:
...statement block...
CASEOF Selector2:
...statement block...
CONTINUE
CASEOF Selector3:
...statement block...
DEFAULT:
...statement block...
ENDSWITCH
```

Explanation: If *Test* matches *Selector1*, execute the statement block after CASEOF *Selector1*. A statement block can call a subroutine (see *Subroutines* below). If CONTINUE follows statement block, the next statement block is automatically executed. See *SWITCH* in *Programming Commands Reference*.

Loop Statements

Execute a statement block a specified number of times, until an expression is true, or while an expression is true. Macros support the following loop statements:

FOR-ENDFOR

Executes a statement block a specified number of times. The general form is,

```
FOR(ControlVariable; InitialValue; TerminateExp; IncrementExp)
...statement block...
ENDFOR
```

Explanation: *InitialValue* initializes *ControlVariable*. *TerminateExp* tests the value of *ControlVariable*. *IncrementExp* increases the value of *ControlVariable* until *TerminateExp* is true and the loop ends.

```
FOR(x; 1; x < 5; x + 1)
...statement block...
ENDFOR
```

Explanation: x is initialized to 1, statement block executes while x is less than 5, x is incremented by 1 at the end of each loop. Similar loop statements are FOREACH-ENDFOR and FORNEXT-ENDFOR. See *Relational Expressions in Variables, Constants, Operators, and Expressions*, and *FOR, FOREACH, and FORNEXT* in *Programming Commands Reference*.

REPEAT-UNTIL

Executes a statement block until the expression result equals True. All REPEAT statements execute at least once because the expression is evaluated at the end of the loop.

```
REPEAT
...statement block...
UNTIL(x >= 10)
```

Explanation: Execute *statement block* until the result of expression $x \geq 10$ equals True (until x is greater than or equal to 10). The value of x must change in the loop to make the expression true or the loop never ends. For example,

```
REPEAT
...statement block...
x := x + 1
UNTIL(x >= 10)
```

Explanation: Increment x by 1 at the end of each loop until x is greater than or equal to 10.

See *Relational Expressions in Variables, Constants, Operators, and Expressions*, and *REPEAT* in *Programming Commands Reference*.

WHILE-ENDWHILE

WHILE statements do not execute unless the expression is initially true because it is evaluated at the start of the loop. For example,

```
WHILE(x <= 10)
...statement block...
ENDWHILE
```

Explanation: Execute statement block while the result of expression $x \leq 10$ equals True (while x is less than or equal to 10). If x is greater than 10, the loop does not execute. The value of x must change in the loop to make the expression true or the loop never ends. For example,

```
WHILE(x <= 10)
...statement block...
x := x + 1
ENDWHILE
```

Explanation: Increment x by 1 at the end of each loop while x is less than or equal to 10.

See *Relational Expressions* in *Variables, Constants, Operators, and Expressions*, and *WHILE* in *Programming Commands Reference*.

Calling Statements

Call a subroutine which consists of one or more statements, such as command or assignment statements (see *Subroutines* below). Macros support the following calling statements:

CALL

Has one parameter which is the name of a subroutine to call. RETURN directs macro execution to the statement that follows CALL (see *CALL* and *RUN* in *Programming Commands Reference*).

GO

Has one parameter which is the name of a subroutine to jump to. Macro execution continues from the point of the subroutine and does not return (statements between GO and the subroutine do not execute). RETURN ends a macro or directs macro execution to the statement that follows a RUN command (see *GO* and *RUN* in *Programming Commands Reference*).

Subroutine Name

Performs the same action as a CALL statement. For example,

```
CALL StartMacro(Parameter1; Parameter2)
StartMacro(Parameter1; Parameter2)
```

both call a function or procedure named StartMacro (see *Subroutines* below). If the second example calls a function, you can assign the return value to a variable with a statement such as

```
x := StartMacro(Parameter1; Parameter2)
```

You can evaluate the result of an expression in loop and conditional statements. For example,

```
APPLICATION (A1; "WordPerfect"; Default; "US")
FUNCTION Add(x)
x := x + 2
RETURN(x)
ENDFUNC
```

```
IF(Add(3) = 5)
BEEP
ENDIF
```

Explanation: The computer beeps because the expression result equals True (5 = 5).

Subroutines

Consist of one or more statements that execute when the subroutine is called by CALL, GO, or the subroutine name. Subroutine names cannot be one of the reserved words (see *Appendix C*).

Macros support the following subroutines:

Labels

Have no parameters. Label names begin with a character, consist of one or more letters or numbers, are limited to 30 characters, and have an optional trailing @ sign. Labels generally include one or more statements followed by RETURN or QUIT. LABEL statements can execute the same as other macro statements, and do not have to be called. If a LABEL is a macro's first executable statement, the same LABEL can be called later in the macro. See LABEL, and the example for SUBSTR in *Programming Commands Reference*.

INDIRECT can create a LABEL name out of a combination of character strings and/or numbers (see INDIRECT in *Programming Commands Reference*). You can use INDIRECT to call a LABEL, but not to create one.

Functions

Can have zero parameters, or one or more parameters which receive a value from a calling statement (see *Calling Statements* above), and return a value. A function without parameters performs like LABEL except that a function can not execute unless called. Function names must begin with a character, consist of one or more letters or numbers, are limited to 30 characters, and have an optional trailing @ sign. See FUNCTION in *Programming Commands Reference*. The following rules apply to functions:

- ♦ Functions begin with the word FUNCTION and end with ENDFUNC.
- ♦ Functions can not execute unless called.
- ♦ Calling statements consist of a function name, which can be followed by one or more parameter values to pass to the function. If there are no parameters, empty parentheses must follow the function name.
- ♦ The number of parameters in a calling statement must match the number of function parameters.
- ♦ A function cannot be defined inside another subroutine.
- ♦ Functions can be placed anywhere in a macro, or in a macro library file (see USE in *Programming Commands Reference*).
- ♦ Functions can be called from a subroutine, or a function can call itself (functions are recursive).
- ♦ Function variables are private to the function. A variable with the same name as a function variable can be used elsewhere in the macro without conflict.

- ♦ Functions can include LABEL statements which are not visible outside the function. A LABEL statement inside a function should *not* have the same name as a function or procedure.
- ♦ Functions accept RETURN statements with no parameters (return 0); or a value contained in a variable that is the result of a function operation; or an enumerated type that asserts a Cancel, Error, or Not Found condition; or an enumerated type that asserts a Cancel, Error, or Not Found condition, and a value contained in a variable that is the result of a function operation See *RETURN* in *Programming Commands Reference*.
- ♦ Function names cannot be one of the reserved words (see *Appendix C*).

Procedures

Can have no parameters, or one or more parameters which receive a value from a calling statement. A procedure cannot return a value. A procedure without parameters performs like LABEL, except it cannot execute unless called. Procedure names begin with a character, consist of one or more letters or numbers, are limited to 30 characters, and have an optional trailing @ sign. See *PROCEDURE* in *Programming Commands Reference*. The following rules apply to procedures:

- ♦ Procedures begin with the word PROCEDURE and end with ENDPROC.
- ♦ Procedures cannot execute unless called.
- ♦ Calling statements consist of a procedure name, which can be followed by one or more parameter values that are passed to the procedure. If there are no parameters, empty parentheses must follow the procedure name.
- ♦ The number of parameters in a calling statement must match the number of procedure parameters.
- ♦ A procedure cannot be defined inside another subroutine.
- ♦ Procedures can be placed anywhere in a macro, or in a macro library file (see *USE* in *Programming Commands Reference*).
- ♦ Procedures can be called from a subroutine, or a procedure can call itself (procedures are recursive).
- ♦ Procedure variables are private to the procedure. A variable with the same name as a procedure variable can be used elsewhere in the macro without conflict.
- ♦ Procedures can include LABEL statements which are not visible outside the procedure. A LABEL statement inside a procedure should *not* have the same name as a procedure or function.
- ♦ Procedures accept RETURN statements with no parameters (directs macro execution to the statement that follows a procedure's caller); or one parameter (an enumerated type that asserts a Cancel, Error, or Not Found condition. See *RETURN* in *Programming Commands Reference*).
- ♦ Procedure names cannot be one of the reserved words (see *Appendix C*).

Macro File Libraries

Contain functions and/or procedures that can be called from another macro, and must be compiled.

USE

Specifies a macro file that contains functions and/or procedures. USE usually precedes calling statements to a macro file library. The following macro library example contains only two functions:

```
FUNCTION Add(x)
  x := x + 50
  RETURN(x)
ENDFUNC
```

```
FUNCTION Subtract(x)
  x:= x - 25
  RETURN(x)
ENDFUNC
```

If the previous example was saved and compiled as LIBRARY.WCM, the next example includes statements that call the Add and Subtract functions:

```
APPLICATION(A1; "WordPerfect"; Default; "US")
USE("C:\WPWIN60\MACROS\LIBRARY.WCM")
z := Add(50)
z := Subtract(z)
IF(z = 75)
  BEEP
ENDIF
```

Explanation: The computer beeps because z equals 75.

The previous example with calling statements in shorthand notation:

```
APPLICATION(WP; "WordPerfect"; Default; "US")
USE("C:\WPWIN60\MACROS\LIBRARY.WCM")
z := Add(Subtract(50))
IF(z = 75)
  BEEP
ENDIF
```


Creating Macros



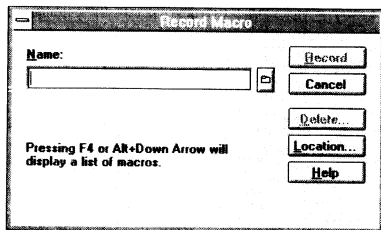
Recording Macros

Recording a macro saves a series of keystrokes and/or mouse actions as commands to play (repeat) later (see *Macro Facility*). Recorded macros automate routine tasks, such as typing the complimentary closing of a letter, displaying a dialog box for user input, or opening or closing a document.

Record a Macro

To record a macro,

- 1 Choose **Tools, Macro, Record** (or press **Ctrl+F10**) to display the Record Macro dialog box.



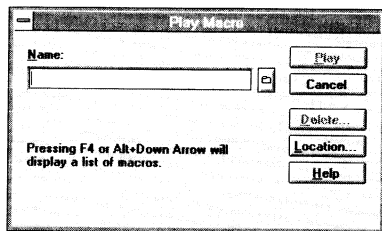
- 2 Enter a filename in the edit box. The default extension **.WCM** is automatically added when you choose **Record** unless you enter a different extension. Enter a path if different from the default macros directory. To insert a macro filename, choose the button to the right of the edit box to display the **Select File** dialog box, then double-click the desired filename.

To change the default extension, start or switch to WordPerfect Macro Facility (see *Macro Facility*). From the Macro Facility main menu, choose **Settings, Options**, then enter the file extension in the **Extension** edit box.

- 3 Choose **Record** to dismiss the dialog box. The insertion point changes to a circle with a slash to indicate that keystrokes and/or mouse actions are being recorded. "Macro Record" appears on the status bar.
- 4 Enter desired text. For example, type **Hello**. Later, when you play the macro, Hello automatically appears in your document.
- 5 Choose **Tools, Macro, Record** to stop recording. The macro is saved in the default macros directory unless you specified a different path (step 2).

Play a Macro

- 1 Choose **Tools, Macro, Play** (or press **Alt+F10**) to display the Play Macro dialog box.

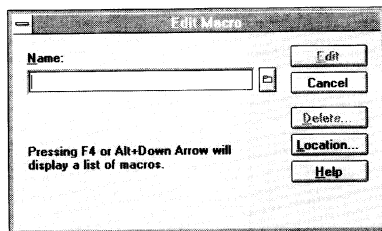


- 2 Enter a filename, including path and extension if necessary, then choose **Play**. To insert a macro filename, choose the button to the right of the edit box to display the **Select File** dialog box, then double-click the desired filename.
- 3 Choose **Play** to dismiss the dialog box and play the macro. Macros automatically compile the first time they are played (see *Macro Facility*).

Edit a Macro

To edit a macro,

- 1 Choose **Tools, Macro, Edit** to display the **Edit Macro** dialog box.



- 2 Enter a filename, including path and extension if necessary, then choose **Edit**. To insert a filename, choose the button to the right of the edit box to display the **Select File** dialog box, then double-click the desired filename.
- 3 Choose **Edit** to dismiss the dialog box, display the Macro Feature Bar, and open the macro. See *Macro Feature Bar* in *Writing Macros*.

If you record the keystrokes Hello as a macro (see *Record a Macro* above), then open the macro file, you see:

```
Application (A1; "WordPerfect"; Default; "US")
Type (Text: "Hello")
```

The macro consists of an **APPLICATION** programming command, and a **Type** product command (see *APPLICATION* in *Programming Commands Reference* and *Type* in *Product Commands Reference*).

- 4 Edit the macro. For example, enter commands that bold and center text between **APPLICATION** and **Type**, and a command that turns bold off after **Type**:

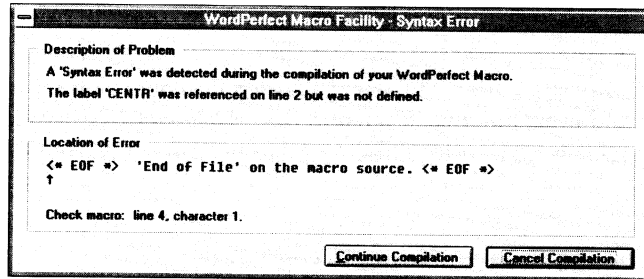
```
Application (A1; "WordPerfect"; Default; "US")
Center ()
AttributeAppearanceOn (Attrib: Bold!)
Type (Text: "Hello")
AttributeAppearanceOff (Attrib: Bold!)
```

See *Center*, *AttributeAppearanceOn*, and *AttributeAppearanceOff* in *Product Commands Reference*. Play the macro to see the changes.

Macro Errors

You can introduce errors when you write or edit a macro (see *Syntax* in *Macro Language Concepts*). If in the example above you typed *Centr* instead of *Center*, when

you choose **Save & Compile**, a compile-time syntax error displays the following syntax error message box:

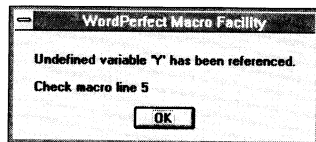


The compiler does not recognize `Centr`, and guesses that it is the calling statement of a LABEL subroutine (see *Calling Statements and Subroutines in Macro Control Statements*). When it reaches the end of the file and does not find a corresponding LABEL, it displays a syntax error message. Common syntax errors are,

- ♦ Missing semicolons between parameters
- ♦ Missing parentheses
- ♦ Missing double quotation marks (see *Character Expressions in Variables, Constants, Operators, and Expressions*)
- ♦ Missing command (often the closing statement) in a conditional or loop statement (see *Conditional Statements and Loop Statements in Macro Control Statements*)
- ♦ Misspelled macro command names
- ♦ Undefined calling statements (see *Calling Statements and Subroutines in Macro Control Statements*)

The compiler identifies syntax errors and suggests solutions for the writer to verify.

Run-time error messages identify errors that occur while the macro plays and where the errors occur in the macro. For example, if you reference a variable not assigned a value, a message box similar to the following is displayed:



To correct the error, initialize variable `Y` (assign `Y` a value) before line 5.

Writing Macros

A recorded macro includes only product commands. When you play a recorded macro, each action is executed in recorded order (see *Recording Macros*).

A written macro can also include non-recordable statements that specify how a macro performs a task. The writer specifies the conditions for executing assignments, loops, and programming commands (see *Macro Language Concepts*). Writing a macro provides greater flexibility to determine what a macro does than recording a macro.

Macro Feature Bar

To write a macro,

- 1 Open a new WordPerfect document.
- 2 Choose **Tools, Macro, Macro Bar** to display the Macro feature bar.



- 3 Enter macro statements by typing them or by using the Command Inserter (see below).
- 4 Choose the **Save & Compile** button. The default file extension (.WCM) is automatically added. To change the default extension, start or switch to WordPerfect Macro Facility (see *Macro Facility*). From the Macro Facility main menu, choose **Settings, Options**, then enter the file extension in the **Extension** edit box.
- 5 Choose the **Play** button (grayed until the macro is saved and compiled).

The following macro example streamlines the start up process for writing a new macro. Use as is, or modify to fit your own needs.

```
//*****  
//*   MACRO: MACWRITE.WCM  
//*   PURPOSE: Start up process for writing a macro  
//*****  
  
APPLICATION(WP; "WordPerfect"; Default; "US")  
ONERROR(Stop)  
  
CALL(Font)  
CALL(Margins)  
CALL(Setup)  
QUIT  
  
LABEL(Font)  
  Display(On!)  
  FileNew  
  Font(Name: "Arial Regular"; Family: FamilyHelvetica!; Attributes: FontMatchNormal!; Weight: 90;  
  Width: WidthUnknown!; Source: DRSFile!; Type: TrueType!; CharacterSet: FontMatchASCII!)  
  FontSize(11p)  
RETURN
```

```

LABEL(Margins)
  MarginTop(0.5")
  MarginBottom(0.5")
  MarginLeft(0.5")
  MarginRight(0.5")
RETURN

```

```

LABEL(Setup)
  MacroEditControlBar
  ViewDraft
  ZoomToMarginWidth
  Type("APPLICATION(WP; ""WordPerfect""; Default; ""US"")")
  HardReturn
  Type("// Purpose: ")
  FileSaveAsDlg
RETURN

```

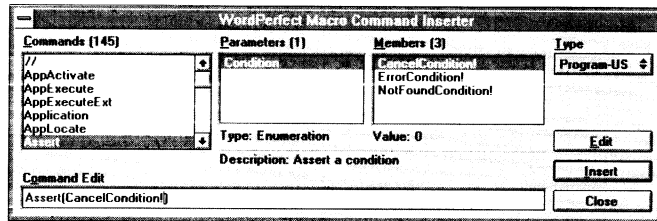
```

LABEL(Stop)
  BEEP
  MESSAGEBOX(x;"WARNING";"Did you save the macro?"; IconExclamation!)
  QUIT

```

Command Inserter

Inserts product and programming commands into a WordPerfect macro document. From the Command Inserter dialog box,



choose the **Type** popup button to select programming commands, Macro Facility product commands, or WordPerfect product commands. Programming commands and Macro Facility product commands can be used in other Shared Code applications. WordPerfect product commands are specific to WordPerfect.

Double-click a command and/or member name (or select a name, then choose **Edit**) to insert the command in the **Command Edit** box. Choose **Insert** to insert the command name into a document.

The following macro example activates Command Inserter from a WordPerfect document:

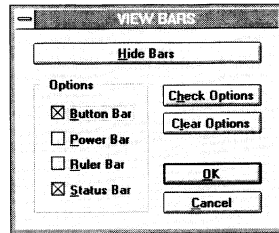
```

//*****
//*   MACRO: CTRL.M.WCM
//*   PURPOSE: Activates the macro command inserter.
//*****
Application (A1; "WordPerfect"; Default; "US")
MacroCommandInserter(On!)

```

Sample Macro

VIEWBARS.WCM displays the View Bars dialog box, where a user selects Bars to show. When the macro first plays, checked boxes indicate selected Bars. The user selects Check Options to show the four main Bars, or Clear Options to hide them. Hide Bars hides all WordPerfect Bars, including the Menu Bar. When the user selects OK, only bars with checked boxes are shown.



Main Sections

VIEWBARS.WCM is written in five sections to help you follow its design and understand how it works.

- ♦ Header
- ♦ Application statement
- ♦ Main Call statements
- ♦ Main Label statements
- ♦ Secondary Label statements

Header

Provides information about the author, name of the macro, directory, and purpose in comment statements.

```
// Author: John Doe
// Name: VIEWBARS.WCM
// Directory: C:\WPWIN60\MACROS
// Purpose: Select bars to show or hide
```

Application Statement

Specifies the application used in the macro.

```
APPLICATION (WP; "WordPerfect"; Default; "US")
```

If your macro uses more than one application, it must include an APPLICATION statement for each one. There is only one default application. See *APPLICATION* in *Programming Commands Reference*.

Main Call Statements

Define a macro's main subroutines (see *Subroutines* in *Macro Control Statements*). Each CALL parameter is the name of a LABEL. The LABEL name should describe the LABEL's purpose. When LABEL finishes (returns), macro execution continues to the next CALL statement, and so forth. The last LABEL ends the macro. Comment statements identify this section as Main Call Statements.

```

//*****
// Main Call Statements
//*****

CALL(InitializeBarVariables)
CALL(CreateDialogBox)
CALL(GetUserChoices)
CALL(ShowBars)
CALL(QuitMacro)

```

Main Label Statements

Perform the main macro tasks. Comment statements identify this section as Labels that correspond to the Main Call Statements.

```

//*****
// MAIN LABEL STATEMENTS
//*****

```

The parameter name of each LABEL statement matches the name of its caller.

LABEL(InitializeBarVariables) ♦ Assigns the current status of Bar system variables to user-defined variables vBB, vPB, vRB, and vSB. The value equals True if active, False if not. The lowercase v in variable names is an optional convention that stands for variable. The indentations are an optional convention to make the macro easier to read. RETURN directs macro execution to CALL(CreateDialogBox).

```

LABEL(InitializeBarVariables)
  vBB := ?ButtonBarActive
  vPB := ?PowerBarActive
  vRB := ?RulerBarActive
  vSB := ?StatusBarActive
RETURN

```

LABEL(CreateDialogBox) ♦ Creates a dialog box with ten controls in memory, but does not display the dialog box. RETURN directs macro execution to CALL(GetUserChoices). See *Dialog Boxes*.

```

LABEL(CreateDialogBox)
  DIALOGDEFINE("ViewBars"; 50; 50; 137; 137; 16; "VIEW BARS")
  DIALOGADDPUSHBUTTON("ViewBars"; 1; 10; 8; 112; 13; 0; "&Hide All Bars")
  DIALOGADDGROUPBOX("ViewBars"; 2; 10; 28; 65; 82; "Options")
  DIALOGADDCHECKBOX("ViewBars"; 3; 20; 45; 50; 10; "&Button Bar"; vBB)
  DIALOGADDCHECKBOX("ViewBars"; 4; 20; 60; 50; 10; "&Power Bar"; vPB)
  DIALOGADDCHECKBOX("ViewBars"; 5; 20; 75; 50; 10; "&Ruler Bar"; vRB)
  DIALOGADDCHECKBOX("ViewBars"; 6; 20; 90; 50; 10; "&Status Bar"; vSB)
  DIALOGADDPUSHBUTTON("ViewBars"; 7; 82; 33; 40; 13; 0; "C&heck All")
  DIALOGADDPUSHBUTTON("ViewBars"; 8; 82; 50; 40; 13; 0; "C&lear All")
  DIALOGADDPUSHBUTTON("ViewBars"; 9; 82; 80; 40; 13; 1; "&OK")
  DIALOGADDPUSHBUTTON("ViewBars"; 10; 82; 97; 40; 13; 0; "&Cancel")
RETURN

```

LABEL(GetUserChoices) ♦ Displays the dialog box and waits for user input. Check boxes are displayed checked if the value of its corresponding variable equals

True. If Hide Bars is chosen, 1 is assigned to MacroDialogResult and LABEL(HideBars) is called. If Check Options is chosen, 7 is assigned and LABEL(CheckOptions) is called. If Clear Options is chosen, 8 is assigned and LABEL(ClearOptions) is called. If OK is chosen, 9 is assigned and RETURN directs macro execution to CALL>ShowBars). If Cancel is chosen, 10 is assigned and LABEL(QuitMacro) is called. For information about implicit variable MacroDialogResult, see *DIALOGDEFINE* in *Programming Commands Reference*.

```

LABEL(GetUserChoices)
    DIALOGDISPLAY("ViewBars"; 9)
    SWITCH(MacroDialogResult)
        CASEOF 1: CALL(HideBars)
        CASEOF 7: CALL(CheckOptions)
        CASEOF 8: CALL(ClearOptions)
        CASEOF 9: RETURN
        CASEOF 10: GO(QuitMacro)
    ENDSWITCH
RETURN

```

LABEL>ShowBars) ♦ Shows or hides a bar depending on the value of vBB, vPB, vRB, and vSB. A bar is shown if the value of the corresponding variable equals True, or hidden if it equals False. RETURN directs macro execution to CALL(QuitMacro).

```

LABEL>ShowBars)
    ButtonBarShow(vBB)
    PowerBarShow(vPB)
    RulerBarShow(vRB)
    StatusBarShow(vSB)
RETURN

```

LABEL(QuitMacro) ♦ Removes the dialog box from memory and ends the macro.

```

LABEL(QuitMacro)
    DIALOGDESTROY("ViewBars")
QUIT

```

Secondary Label Statements

Perform secondary macro tasks that support the main task. If the main task is to display a list of options, the secondary task is to execute a statement block depending on which option the user chooses. A secondary LABEL statement is called from a main LABEL statement. The parameter name of each LABEL command matches the parameter name of its caller (see *Main Call Statements* above). Comment statements identify this section as Secondary Label Statements.

```

//*****
// SECONDARY LABEL STATEMENTS
//*****

```

LABEL(HideBars) ♦ Calls HideBars product command and LABEL(QuitMacro) to end macro. Called by CALL(HideBars) in LABEL(GetUserChoices).


```
LABEL(HideBars)
  HideBars
  CALL(QuitMacro)
```

LABEL(CheckOptions) ♦ Sets user-defined Bar variables to True (see *Main Label Statements* above). When the dialog box is displayed again, the check boxes are checked. GO(GetUserChoices) creates a loop between the dialog box and this Label statement. Called by CALL(CheckOptions) in LABEL(GetUserChoices).

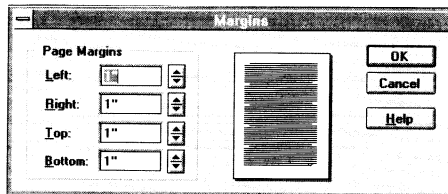
```
LABEL(CheckOptions)
  vBB := True
  vPB := True
  vRB := True
  vSB := True
  GO(GetUserChoices)
```

LABEL(ClearOptions) ♦ Sets user-defined Bar variables to False (see *Main Label Statements* above). When the dialog box is redisplayed, the check boxes are blank. GO(GetUserChoices) creates a loop between the dialog box and this Label statement. Called by CALL(ClearOptions) in LABEL(GetUserChoices).

```
LABEL(ClearOptions)
  vBB := False
  vPB := False
  vRB := False
  vSB := False
  GO(GetUserChoices)
```

Dialog Boxes

Users interact with applications through menus and dialog boxes. Dialog boxes have more than one option for user input. For example, in the Margins dialog box, default settings are displayed in edit boxes next to spin buttons. To display the Margins dialog box, Choose **Layout, Margins** (an ellipsis to the right of a menu option indicates a dialog box is available).



To change settings, enter new values in the edit box, or click a spin button. The edit box and spin button are Windows controls for user input. Controls available to a dialog box are described below.

Dialog Box Controls

Input or output windows where the user interacts with a dialog box and its parent application. WordPerfect supports the following dialog box controls (see *Programming Commands Reference* for control command descriptions):

Check Boxes

Display one or more options. Use a callback function to activate user-defined responses. See *DIALOGDISPLAY* and *DIALOGADDCHECKBOX*.

Color Wheels

Display colors to select. See *DIALOGADDCOLORWHEEL*.

Combination Boxes

Display an edit box and a list box. Enter text in the edit box, or double-click a list item to insert. See *DIALOGADDCOMBOBOX*.

Counter Buttons

Display an edit box and a counter button. Enter a number in the edit box, or click the counter button to insert a number. See *DIALOGADDCOUNTER*.

Edit Boxes

Receive text input. There are different styles of edit controls, including single and multiple line. See *DIALOGADDEDITBOX*.

Filename Edit Boxes

Display an edit control and a Directory button control. Enter a filename in the edit control, or click the Directory button to display the Select File dialog box. See *DIALOGADDFILENAMEBOX*.

Frames

Group dialog box controls. Accept no input. See *DIALOGADDFRAME*.

Group Boxes

Group dialog box controls, and include a title in the frame. Accept no input. See *DIALOGADDGROUPBOX*.

Horizontal Lines

Separate items in a dialog box. Accept no input. See *DIALOGADDHLINE*.

Hot Spots

Are invisible controls that close a dialog box when the user clicks a defined area. Redefine the response with a callback function. See *DIALOGDISPLAY* and *DIALOGADDHOTSPOT*.

Icons

Display graphic representations. Accept no input, unless used with a callback function and *DIALOGADDHOTSPOT*. See *DIALOGDISPLAY*, *DIALOGADDHOTSPOT*, and *DIALOGADDICON*.

List Boxes

Display a list of options to choose from. There are different styles of list boxes, including single and multicolumn. See *DIALOGADDLISTBOX*.

Popup Buttons

Display the name of a menu item on a push button. Click the button to display the menu. See *DIALOGADDPUPUPBUTTON*.

Push Buttons

Activate user-defined responses when chosen. See *DIALOGADDPUSHBUTTON*.

Radio Buttons

Display mutually-exclusive options. Use a callback function to activate user-defined responses. See *DIALOGDISPLAY* and *DIALOGADDRADIOBUTTON*.

Scroll Bars

Scroll through documents, or activate user-defined responses with a callback function. See *DIALOGDISPLAY* and *DIALOGADDSCROLLBAR*.

Static Text Controls

Send text to a dialog box. Accept no input. See *DIALOGADDTEXT*.

Viewers

Display read-only files. See *DIALOGADDVIEWER*.

Vertical Lines

Separate items in a dialog box. Accept no input. See *DIALOGADDVLINE*.

Modal Dialog Boxes

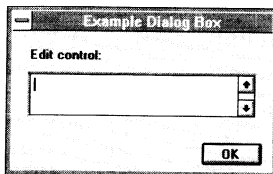
Require the user to complete an action before input to the parent application is allowed. For example, the Margins dialog box requires the user to choose OK or Cancel before control returns to WordPerfect. Macro dialog boxes are modal by default (see *DIALOGDEFINE* in *Programming Commands Reference*).

Modeless Dialog Boxes

Do not require the user to complete an action before input to the parent application is allowed. For example, you can perform a search with the Find Text dialog box (choose Edit, Find), then click and edit the document while the Find Text dialog box remains on the screen. Click the dialog box to perform another search. For creating a modeless dialog box, see *DIALOGDEFINE* in *Programming Commands Reference*.

Creating a Dialog Box

To create a dialog box, define it, add controls, and display it. You must destroy (remove from memory) the dialog box when the macro ends. The commands presented below create the following dialog box:



Define

Specify the size and style of the dialog box, and the controls it will contain. The following command

```
DIALOGDEFINE(Dialog: 1000; Left: 50; Top: 50; Width: 200; Height: 100; Style: OK! |Percent!;  
Caption: "Example Dialog Box")
```

defines (creates in memory) a dialog box with the following specifications:

Dialog • The number 1000 was chosen to identify the dialog box. Names such as "MainDlg" or "ViewBarsDlg" are also valid. The number or name must be unique. It is used again to add controls, display, and then destroy the dialog box when the macro ends.

Left, Top, Width, Height • The dialog box is displayed in the center of the screen by setting the left and top parameters to 50 and using the Percent! style. Dialog boxes are positioned and sized in dialog units. A vertical unit equals 1/8 the font height, and a horizontal unit equals 1/4 the font width.

Style • The Percent! formula is

*(screen width or height - dialog box width or height) * percentage*

It equals the number of dialog units from the left side of the screen to the left side of the dialog box, or the number of dialog units from the top of the screen to the top of the dialog box. The dialog box has an OK push button control.

Caption • The dialog box caption is "Example Dialog Box".

Parameter Names • Parameter names are optional. For example,

```
DIALOGDEFINE(1000; 50; 50; 200; 125; OK! |Percent!; "Example Dialog Box")
```

defines the same dialog box as the command above.

Add Controls

The following commands

```
DIALOGADDTTEXT(Dialog: 1000; Control: 100; Left: 10; Top: 10; Width: 50; Height: 15; Style: Left!;  
Text: "Edit control:")
```

```
DIALOGADDEDITBOX(Dialog: 1000; Control: 101; Left: 10; Top: 25; Width: 175; Height: 50; Style:  
Left! |VScroll! |Multiline!; MacroVar: vReturn; LimitText: 1000)
```

add a static control and an edit control to the dialog box with the following specifications:

Dialog • 1000 for both controls, which equals the *Dialog* parameter of the parent dialog box (see *Define* above).

Control • The controls have unique Control numbers (100 and 101), which could also be names such as "Control 1" and "Control 2". DIALOGDISPLAY uses this number or name to give a control the initial input focus.

The value of the *Control* used to dismiss the dialog box is returned in the implicit variable MacroDialogResult (see *DIALOGDEFINE* in *Programming Commands*

Reference): 1 for OK, 2 for Cancel, 2 for Close (system menu box), 2 if you double-click the system menu box, 2 if you press Alt+F4, or the value of the *Control* parameter of a user-defined push button or hot spot (see *DIALOGADDPUSHBUTTON* and *DIALOGADDHOTSPOT* in *Programming Commands Reference*).

Left, Top, Width, Height • The controls are displayed 10 dialog units from the left side of the dialog box, and 10 and 25 units from the top. The widths are 50 and 175 dialog units, and the heights are 15 and 50 dialog units. A vertical dialog unit equals 1/8 the font height, and a horizontal dialog unit equals 1/4 the font width.

Style • The static text is left justified. The edit box is multi-line with a vertical scroll bar. Combine styles with bitwise OR operator (`|`) (see *Bitwise Operators* in *Variables, Constants, Operators, and Expressions*).

Text • The static control text is "Edit Control".

MacroVar, LimitText • Text typed into the edit box is assigned to variable `vReturn` when the dialog box is dismissed by OK or Close on the system menu. The maximum number of characters the edit control accepts is 1000.

Display

The following command

```
DIALOGDISPLAY(1000; 101)
```

displays dialog box 1000 and gives the input focus to control 101 (edit control has the insertion point).

Destroy

The following command

```
DIALOGDESTROY(1000)
```

removes dialog box 1000 from memory, including the value of the implicit variable `MacroDialogResult`. If you need this value, assign it to a another variable before executing `DIALOGDESTROY`. If you do not destroy a dialog box before the macro ends, memory conflicts may occur.

Macro Examples

The following sections, *Macro Menus*, *Creating Callback Functions*, and *Dynamic Data Exchange* present four macro examples to demonstrate calling statements, conditional statements, loop statements, subroutines, and macro techniques such as callback functions. The examples have no practical purpose except to show techniques that the user can use to write macros.

Copy the examples from the Online Macros Manual (choose **Help, Macros, Macro Examples**) and then play them in WordPerfect 6.0 for Windows (choose **Tools, Macros, Play**). You may have to edit long lines to remove misplaced hard returns that Clipboard inserts. If you type the examples, do not include line numbers. They reference the line-by-line descriptions included in this manual.

The following macro shows the general form used in the examples:

```
1  APPLICATION(WP; "WordPerfect"; Default; "US")
2  // Name: TEST.WCM
3  // Directory: C:\WP\WIN60\MACROS
4  // Description: Demonstrate PROCEDURE and GETNUMBER commands

5  PROCEDURE ErrorMsg(x)
6      BEEP BEEP BEEP
7      IF(x < 1)
8          vMessage := x + " is less than 1"
9      ELSE
10         Vmessage := x + " is greater than 5"
11     ENDIF
12     MESSAGEBOX(vStatus; "OUT-OF-RANGE ERROR" ; Vmessage + " - please try again";
13         IconExclamation!)
14 ENDPROC

14 REPEAT
15     GETNUMBER(vNum; "Enter a number from 1 to 5"; "GETNUMBER EXAMPLE")
16     IF((vNum < 1) OR (vNum > 5))
17         CALL ErrorMsg(vNum)
18     ELSE
19         MESSAGEBOX(vStatus; "YES!"; "You entered " + vNum; IconInformation!)
20     ENDIF
21 UNTIL((vNum > 0) AND (vNum < 6))
```

Line Description

- | | |
|-----|---|
| 1 | WordPerfect is the default application. The product prefix WP is user-defined in written macros (see <i>APPLICATION</i> in <i>Programming Commands Reference</i>). |
| 2-4 | Comments about the macro. |
| 3 | Macros are generally saved to the default macros directory. |

Line	Description
5-13	Procedure <code>ErrorMsg</code> is called from line 17 when the user enters a number less than 1 or greater than 5. Since Procedures and Functions must be called to execute, their placement does not affect macro execution.
6	The computer beeps three times. You can place macro commands on the same line.
7-11	An IF statement tests the value of <code>x</code> , which is the value the user enters (line 15) that is passed to Procedure <code>ErrorMsg</code> (line 17).
7	If the value of <code>x</code> is less than 1, execute line 8 (assign character string to variable <code>vMessage</code>).
9	If the value of <code>x</code> is greater than 1, execute line 10 (assign character string to variable <code>vMessage</code>).
11	ENDIF closes the IF statement (line 7).
12	MESSAGEBOX displays an out-of-range error (the value of <code>x</code> is less than 1 or greater than 5).
13	ENDPROC closes the PROCEDURE statement (line 5).
14-21	Macro begins with a REPEAT statement that prompts the user to enter a number until the number is in the range 1 to 5.
15	GETNUMBER prompts the user to enter a number from 1 to 5.
16-20	An IF statement tests the value of variable <code>vNum</code> , which is the value the user entered (line 15). If <code>vNum</code> is less than 1 or greater than 5 call procedure <code>ErrorMsg</code> (line 17).
18	If variable <code>vNum</code> is in the range 1 to 5, display a message that shows the number entered by the user (line 19).
20	ENDIF closes the IF statement (line 16).
21	UNTIL closes the REPEAT loop statement (line 14). The loop continues until variable <code>vNum</code> is greater than 0 and less than 6 (in the range 1 to 5).

Macro Menus

The following macro uses MENU to display a list of options, ASSERT to create a Cancel, Error, or Not Found condition, ONCANCEL, ONERROR, or ONNOTFOUND to execute a subroutine. It also uses MESSAGEBOX to display messages, and GO to create user-defined loops. Line-by-line comments follow the text of the macro. See *Programming Commands Reference* for detailed command descriptions.

```
1 // Name: ASSERT.WCM
2 // Directory: C:\WPWIN60\MACROS
3 // Description: Demonstrate MENU, MESSAGEBOX, ASSERT, ONCANCEL, ONERROR,
  ONNOTFOUND, CANCEL, ERROR, NOTFOUND

4 APPLICATION(WP; "WordPerfect"; Default: "US")

5 CALL(AssertConditions)
6 CALL(DisplayMenu)
7 CALL(QuitMacro)

8 //*****
9 // MAIN LABEL STATEMENTS
10 //*****

11 LABEL(AssertConditions)
12   ONCANCEL(Conditions)
13   ONERROR(Conditions)
14   ONNOTFOUND(Conditions)
15   vCondition := "Turn conditions Off!"
16   RETURN

17 LABEL(DisplayMenu)
18   MENU(vChoice; Digit!;; {"CancelCondition!"; "ErrorCondition!"; "NotFoundCondition!";
  vCondition; "Quit"})
19   SWITCH(vChoice)
20     CASEOF 1: ASSERT(CancelCondition!)
21     CASEOF 2: ASSERT(ErrorCondition!)
22     CASEOF 3: ASSERT(NotFoundCondition!)
23     CASEOF 4: IF(vCondition = "Turn conditions Off!")
24       GO(TurnConditionsOff)
25     ELSE
26       GO(TurnConditionsOn)
27     ENDIF
28     CASEOF 5: GO(QuitMacro)
29     DEFAULT: Conditions
30   ENDSWITCH
31   RETURN

32 LABEL(QuitMacro)
33   vMessage := "You selected 5, or you selected 1, 2, or 3 with all conditions Off!"
34   MESSAGEBOX(vStatus; "Condition Status"; vMessage; RetryCancel! | IconInformation! |
  DefButton2!)
35   IF(vStatus = 4)
```



```

36     GO(DisplayMenu)
37     ELSE
38     QUIT
39     ENDFIF

40 //*****
41 // SECONDARY LABEL STATEMENTS
42 //*****

43 LABEL(Conditions)
44     SWITCH(ErrorNumber)
45         CASEOF 1: vMessage := "^1"
46         CASEOF 2: vMessage := "^2"
47         CASEOF 7: vMessage := "^3"
48         DEFAULT: vMessage := "^4"
49     ENDSWITCH
50     MESSAGEBOX(vStatus; "Condition Status"; vMessage; Ok! | IconInformation! | HasParameters!;
    {"You selected 1 with CANCEL(On!); "You selected 2 with ERROR(On!); "You selected 3 with
    NOTFOUND(On!); "Menu item not chosen"})
51     GO(DisplayMenu)

52 LABEL(TurnConditionsOn)
53     CANCEL(On!)
54     ERROR(On!)
55     NOTFOUND(On!)
56     vCondition := "Turn conditions Off!"
57     GO(DisplayMenu)

58 LABEL(TurnConditionsOff)
59     CANCEL(Off!)
60     ERROR(Off!)
61     NOTFOUND(Off!)
62     vCondition := "Turn conditions On!"
63     GO(DisplayMenu)

```

Lines Description

- 1-3 Comments about the macro.
- 2 Macros are generally saved to the application's macros directory.
- 4 WordPerfect identified as the default application.
- 5-7 Comments identify CALL statements that outline and define the macro's basic operations.
- 5 Call LABEL(AssertConditions), which identifies a LABEL subroutine to call when there is a Cancel, Error, or Not Found condition (lines 12-14).
- 6 Call LABEL(DisplayMenu) after LABEL(AssertConditions) returns (line 16). LABEL(DisplayMenu) displays a five-option menu (line 18).
- 7 Call LABEL(QuitMacro) after LABEL(DisplayMenu) returns (line 31). LABEL(QuitMacro) displays a message box where the user can end the macro or redisplay the menu (lines 32-39).

- 8-10 Comments identify LABELs that correspond to the opening CALL statements (lines 5-7).
- 11-16 LABEL(AssertConditions) identifies a LABEL to execute when a Cancel, Error, or Not Found condition occurs.
- 12 Execute LABEL(Conditions) when a Cancel condition occurs (line 43).
- 13 Execute LABEL(Conditions) when an Error condition occurs (line 43).
- 14 Execute LABEL(Conditions) when a Not Found condition occurs (line 43).
- 15 Assignment operator assigns a character string to a variable named vCondition, which is menu item 4 the first time the menu is displayed (line 18). The default condition for CANCEL, ERROR, and NOTFOUND is On!.
- 16 RETURN statement directs macro execution to CALL(DisplayMenu) (line 6).
- 17-31 LABEL displays a five-option menu and asserts a Cancel, Error, or Not Found condition depending on a menu choice.
- 18 MENU displays a numbered (Digit!) list of five items. The number the user chooses is assigned to vChoice. Omitting the third and fourth parameters centers the menu on the screen.
- 19 SWITCH initiates an action, depending on the value of vChoice, which contains the number of the chosen menu item.
- 20 If 1 is selected, assert a Cancel condition. The macro assigns 1 to the variable ErrorNumber (see *ASSERT* in *Programming Commands Reference*). LABEL(Condition) is automatically called (lines 12 and 45).
- 21 If 2 is selected, assert an Error condition. The macro assigns 2 to the variable ErrorNumber (see *ASSERT* in *Programming Commands Reference*). LABEL(Condition) is automatically called (lines 13 and 46).
- 22 If 3 is selected, assert a Not Found condition. The macro assigns 7 to the variable ErrorNumber (see *ASSERT* in *Programming Commands Reference*). LABEL(NotFound) is automatically called (lines 14 and lines 47).
- 23-27 If 4 is selected, toggle CANCEL, ERROR, and NOTFOUND conditions by calling LABEL(TurnConditionsOn) (lines 52-57) or LABEL(TurnConditionsOff) (lines 58-63). The label called depends on the contents of variable vCondition (lines 15, 56, and 62). Update the description of menu item 4.
- 28 If 5 is selected, end (quit) macro.
- 29 If another key is pressed, such as Esc, execute LABEL(Condition). Notice that the subroutine name executes the subroutine. See *Subroutine Name* in *Macro Control Statements*.
- 30 ENDSWITCH closes SWITCH statement (line 19).
- 31 RETURN statement directs macro execution to CALL(QuitMacro).

- 32-39 LABEL(QuitMacro) displays a message box with an option to end the macro or redisplay the menu.
- 33 Assignment operator assigns character string "You selected 5, or you selected 1, 2, or 3 with all conditions Off!" to variable vMessage (line 34).
- 34 MESSAGEBOX displays a message box with an information icon, and Retry and Cancel buttons. If Retry is chosen, 4 is assigned to variable vStatus, or 2 if Cancel is chosen.
- 35-36 If variable vStatus equals 4 (line 34), GO sends macro execution to LABEL(DisplayMenu), which redisplay the menu.
- 37-38 If vStatus is any value except 4 (ELSE), end the macro (QUIT).
- 39 ENDIF closes IF statement (line 35).
- 40-42 Comments identify secondary LABELs or LABELs called from a another LABEL (see lines 12-14, 24, and 26).
- 43-51 Execute LABEL(Conditions) when the user chooses 1, 2, or 3 (lines 20-22 and 12-14).
- 44 SWITCH initiates an action, depending on the value of the variable ErrorNumber (lines 20, 21, 22) (see *ASSERT* in *Programming Commands Reference*).
- 45 If ErrorNumber equals 1 (Cancel condition), assign "^1" to vMessage (see line 50).
- 46 If ErrorNumber equals 2 (Error condition), assign "^2" to vMessage (see line 50).
- 47 If ErrorNumber equals 7 (NotFound condition), assign "^3" to vMessage (see line 50).
- 48 If ErrorNumber contains a value other than 1, 2, or 7, assign "^4" to vMessage (line 50).
- 49 ENDSWITCH closes SWITCH statement (line 44).
- 50 MESSAGEBOX displays a message box with an information icon and OK button. It uses the HasParameters! style, followed by four *Message* parameter options. A caret (^) followed by a number in the *Message* parameter inserts the corresponding *ParameterData* message (one of the four message options) in its place (lines 45-48). Numbering begins with 0. See *MESSAGEBOX* in *Programming Commands Reference*.
- 51 GO creates a loop by sending macro execution to LABEL(DisplayMenu), which redisplay the menu.
- 52-55 LABEL(TurnConditionsOn) turns on CANCEL, ERROR, and NOTFOUND conditions after LABEL(TurnConditionsOff) is executed.
- 56 Assigns character string "Turn conditions Off!" to variable vCondition, which is displayed as menu item 4 (line 18).

- 57 GO creates a loop by sending macro execution to LABEL(DisplayMenu), which redisplay the menu.
- 58-61 LABEL(TurnConditionsOff) turns off CANCEL, ERROR, and NOTFOUND conditions after the macro first starts or LABEL(TurnConditionsOn) is executed.
- 62 Assigns character string "Turn conditions On!" to variable vCondition, which is displayed as menu item 4 (line 18).
- 63 GO statement creates a loop by sending macro execution to LABEL(DisplayMenu), which redisplay the menu.

Creating Callback Functions

The example macros CALLBACK.WCM and CB_SCBAR.WCM demonstrate techniques available to macro callback functions. To play or edit the macros, copy them from the WordPerfect for Windows 6.0a Online Macros Manual.

CALLBACK.WCM

The following macro creates a callback function that responds to system menu box options, check boxes, radio buttons, hot spots, and push buttons. Line-by-line comments follow the macro text. See *Programming Commands Reference* for detailed command descriptions.

```

1 // Name: CALLBACK.WCM
2 // Directory: C:\WPWIN60\MACROS
3 // Description: Callback function displays a message when a user chooses a dialog box control

4 APPLICATION(WP; "WordPerfect"; Default; "US")

5 //*****
6 // MAIN CALL STATEMENTS
7 //*****

8 CALL(InitializeVariables)
9 CALL(CreateDialog)
10 CALL(DisplayMainDialog)
11 CALL(StartMessageLoop)
12 CALL(QuitMacro)

13 //*****
14 // MAIN LABEL STATEMENTS
15 //*****

16 LABEL(InitializeVariables)
17     WM_SYSCOMMAND := 274
18     WM_COMMAND := 273
19     vCheck := 0
20     vRadio := 0
21     vLoop := "Yes"

```

```

22     RETURN

23 LABEL(CreateDialog)
24     DIALOGDEFINE(1000; 50; 50; 210; 175; 1+2+16; "Callback Function")
25     DIALOGADDCHECKBOX(1000; 101; 10; 10; 50; 10; "Checkbox"; vCheck)
26     DIALOGADDRADIOBUTTON(1000; 102; 10; 30; 60; 10;"Radio button"; vRadio)
27     DIALOGADDDHOTSPO(1000; 103; 90; 10; 50; 50; 1)
28     DIALOGADDFRAME(1000; 104; 90; 10; 50; 50; 5)
29     DIALOGADDTTEXT(1000; 105; 150; 30; 50; 15; 1; "<- Hot spot")
30     vText1 := "Choose Close from the system menu, double-click"
31     vText2 := "the system menu box, or press Alt+F4."
32     DIALOGADDTTEXT(1000; 106; 20; 95; 190; 15; 1; vText1)
33     DIALOGADDTTEXT(1000; 107; 20; 105; 170; 15; 1; vText2)
34     DIALOGADDGROUPBOX(1000; 109; 10; 75; 185; 50; "System commands")
35     RETURN

36 LABEL(DisplayMainDialog)
37     DIALOGDISPLAY (1000; 1; MsgLoop)
38     RETURN

39 LABEL(StartMessageLoop)
40     WHILE(vLoop = "Yes")
41         ENDWHILE
42     RETURN

43 LABEL(QuitMacro)
44     DIALOGDESTROY(1000)
45     QUIT
46     //*****
47     // CALLBACK FUNCTION
48     //*****

49 LABEL(MsgLoop)
50     SWITCH(MsgLoop[3])
51         CASEOF 101:
52             DIALOGUNDISPLAY(1000; "OKBttn")
53             MESSAGEBOX(vStatus; "Check Box"; "Check boxes display options. Choose OK to hide
the message box and return to the main dialog box."; OK! | IconInformation!)
54             DIALOGDISPLAY (1000;1; MsgLoop)
55         CASEOF 102:
56             MESSAGEBOX(vStatus; "Radio Button"; "Radio buttons display mutually-exclusive
options. Choose OK to hide the message box and return to the main dialog box."; OK! |
IconInformation!)
57         CASEOF 103:
58             MESSAGEBOX(vStatus; "Hot Spot"; "A hot spot was created over a filled frame. You can
also create a hot spot over an icon. Choose OK to hide the message box and return to the
main dialog box."; OK! | IconInformation!)
59         CASEOF "OKBttn":
60             MESSAGEBOX(vStatus; "OK Button"; "You chose the OK button on the main dialog box.
Choose OK to hide the message box and return to the main dialog box."; OK! |
IconInformation!)
61         CASEOF "CancelBttn":

```

```

62     MESSAGEBOX(vStatus; "Cancel Button"; "Choosing Cancel assigns ""No"" to variable
        vLoop, which ends the loop and the macro."; OK! | IconExclamation!)
63     vLoop := "No"
64     ENDSWITCH

65     IF(MsgLoop[5] = WM_SYSCOMMAND)
66         MESSAGEBOX(vStatus; "System Command"; "After you choose OK, the main dialog box will
            disappear and then reappear after one second."; OK! | IconInformation!)
67         DIALOGUNDISPLAY(1000; "OKBtn")
68         WAIT(10)
69         DIALOGDISPLAY(1000; 1; MsgLoop)
70     ENDIF
71     RETURN

```

Lines Description

- 1-3 Comments about the macro.
- 2 Macros are generally saved to the default macros directory.
- 4 WordPerfect identified as the default application.
- 5-7 Comments identify the opening CALL statements that outline and define the macro's basic operations.
- 8 Call LABEL(InitializeVariables), which initializes all variables (lines 16-22). A run-time error occurs if you reference a variable that has not been assigned a value. See *VARERRCHK* in *Programming Commands Reference*.
- 9 Call LABEL(CreateDialog) after LABEL(InitializeVariables) returns. LABEL(CreateDialog) creates a dialog box in memory (lines 23-35). It does not display the dialog box.
- 10 Call LABEL(DisplayMainDialog) after LABEL(CreateDialog) returns. LABEL(DisplayMainDialog) displays the dialog box and identifies a callback function. See *DIALOGDISPLAY* in *Programming Commands Reference*.
- 11 Call LABEL(StartMessageLoop) after LABEL(DisplayDialog) returns. LABEL(StartMessageLoop) starts a message loop that suspends the dialog box for callback functions (lines 49-71).
- 12 Call LABEL(QuitMacro) after LABEL(StartMessageLoop) returns. LABEL(QuitMacro) removes the dialog box from memory and ends the macro.
- 13-15 Comments identify LABELs that correspond to the opening CALL statements (lines 8-12).
- 16-22 LABEL(InitializeVariables) initializes all variables.
- 17-18 Initialize variables to contain Windows message numbers. Using variables makes the macro easier to read. The variable names are user-defined, but match the Windows message names. See *DIALOGDISPLAY* in *Programming Commands Reference*.
- 19-20 Initialize check box and radio button variables (lines 25, 26).

Lines	Description
21	Initialize the loop variable. <code>vLoop</code> is later assigned a new value to end the message loop (line 63), which sends macro execution to <code>CALL(QuitMacro)</code> .
22	<code>RETURN</code> sends macro execution to <code>CALL(CreateDialog)</code> .
23-35	<code>LABEL(CreateDialog)</code> creates a dialog box in memory.
25-27	<i>Control</i> parameter values are sent as messages to the callback function (52-63).
35	<code>RETURN</code> sends macro execution to <code>CALL(DisplayMainDialog)</code> .
36-38	<code>LABEL(DisplayMainDialog)</code> displays the dialog box and identifies the callback function <code>MsgLoop</code> . <code>RETURN</code> sends macro execution to <code>CALL(StartMessageLoop)</code> .
39-42	<code>LABEL(StartMessageLoop)</code> starts the message loop that suspends the dialog box for callback functions (lines 49-71). <code>RETURN</code> sends macro execution to <code>CALL(QuitMacro)</code> .
43-45	<code>LABEL(QuitMacro)</code> removes the dialog box from memory and ends the macro.
46-48	Comments identify the callback function.
49-71	Callback function <code>MsgLoop</code> responds to five <i>Control</i> parameter messages and one system command message. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> .
50	<code>SWITCH</code> tests the third element of the callback function array. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> .
51-63	Array element <code>MsgLoop[3]</code> contains the <i>Control</i> parameter value of the control that calls the callback function. A message box describes the control. In this macro, <code>Cancel</code> is the only button that can dismiss the dialog box.
52	<code>DIALOGUNDISPLAY</code> hides the dialog box while the message box is displayed (line 54).
54	<code>DIALOGDISPLAY</code> redisplay the dialog box after the message box is dismissed (line 52).
64	<code>ENDSWITCH</code> closes the <code>SWITCH</code> statement (line 50).
65-70	A <code>WM_SYSCOMMAND</code> message is received when the user chooses <code>Close</code> from the system menu box, double-clicks the system menu box, or presses <code>Alt+F4</code> .
67-69	<code>DIALOGUNDISPLAY</code> hides the dialog box after the message box is dismissed. <code>WAIT(10)</code> pauses the macro for one second. <code>DIALOGDISPLAY</code> redisplay the dialog box.
70	<code>ENDIF</code> closes the <code>IF</code> statement (line 65).
71	<code>RETURN</code> sends macro execution to the message loop (lines 39-42).

CB_SCBAR.WCM

The following macro creates a callback function that responds to Windows WM_VSCROLL, WM_ACTIVATE, and WM_SYSCOMMAND messages. It uses DIALOGADDSCROLLBAR to display a list item. Line-by-line comments follow the macro text. See *Programming Commands Reference* for detailed command descriptions.

```
1 // Name: CB_SCBAR.WCM
2 // Directory: C:\WPWIN60\MACROS
3 // Description: Use DIALOGADDSCROLLBAR and a callback function

4 APPLICATION(WP; WordPerfect; Default; "US")

5 //*****
6 // MAIN CALL STATEMENTS
7 //*****

8 CALL(InitializeVariables)
9 CALL(CreateDlg)
10 CALL(DisplayDlg)
11 CALL(StartMessageLoop)
12 CALL(QuitMacro)

13 //*****
14 // MAIN LABEL STATEMENTS
15 //*****

16 LABEL(InitializeVariables)
17   SB_LINEUP := 0
18   SB_LINEDOWN := 1
19   SB_PAGEUP := 2
20   SB_PAGEDOWN := 3
21   SB_THUMBPOSITION := 4
22   SB_THUMBTRACK := 5
23   SB_TOP := 6
24   SB_BOTTOM := 7
25   SB_ENDSCROLL := 8
26   vOldPos := 10
27   y := 1
28   WM_SYSCOMMAND := 274
29   WM_VSCROLL := 277
30   WM_ACTIVATE := 6
31   vLoop := 1
32   DECLARE List[10]
33   vElement := 1
34   FOREACH(vFruit; {"Apples"; "Oranges"; "Papaya"; "Grapes"; "Bananas"; "Pears"; "Watermelon";
35     "Apricots"; "Peaches"; "Plums"})
36     List[vElement] := vFruit
37     vElement := vElement + 1
38   ENDFOR
39   RETURN

39 LABEL(CreateDlg) // define dialog box and controls
40   DIALOGDEFINE(1000; 50; 50; 155; 160; 16; "Scroll Bar Callback Function")
```



```

41     DIALOGADDSCROLLBAR(1000; "Scroll"; 135; 10; 0; 100; 2 + 8; BarPos; 1; 10)
42     FORNEXT(yPos; 10; 100; 10)
43         DIALOGADDTTEXT(1000; 0; 10; yPos; 40; 10; 2; List[yPos / 10])
44     ENDFOR
45     DIALOGADDTTEXT(1000; 0; 60; 10; 40; 10; 1; "Apples")
46     DIALOGADDTTEXT(1000; 0; 50; yPos + 15; 70; 10; 1; "Thumb Position 1")
47     RETURN

48 LABEL(DisplayDlg) // show dialog box
49     DIALOGDISPLAY (1000; 121; MsgLoop) // callback function = MsgLoop
50     RETURN

51 LABEL(StartMessageLoop)
52     WHILE(vLoop = 1) // message loop
53     ENDWHILE
54     RETURN

55 LABEL(QuitMacro)
56     DIALOGDESTROY(1000)
57     QUIT

58 //*****
59 // CALLBACK FUNCTION
60 //*****

61 LABEL(MsgLoop)
62     SWITCH(MsgLoop[5])
63         CASEOF WM_SYSCOMMAND:
64             GO(QuitMacro)
65         CASEOF WM_VSCROLL:
66             CALL(ScrollCtrl)
67         CASEOF WM_ACTIVATE:
68             CALL(MsgBox)
69     ENDSWITCH

70     SWITCH(MsgLoop[3])
71         CASEOF "CancelBtn":
72             GO(QuitMacro)
73     ENDSWITCH
74     RETURN

75 LABEL(ScrollCtrl)
76     SWITCH(MsgLoop[6])
77         CASEOF SB_LINEUP:
78             CALL(LineUp)
79         CASEOF SB_LINEDOWN:
80             CALL(LineDown)
81         CASEOF SB_PAGEUP:
82             CALL(PageUp)
83         CASEOF SB_PAGEDOWN:
84             CALL(PageDown)
85         CASEOF SB_THUMBPOSITION:
86             CALL(Thumb)

```

```

87     //CASEOF SB_THUMBTRACK:
88     // CALL(Thumb)
89     CASEOF SB_TOP :
90         CALL(PageUp)
91     CASEOF SB_BOTTOM:
92         CALL(PageDown)
93     CASEOF SB_ENDSCROLL:
94         DIALOGADDTTEXT(1000; 0; 107; 125; 10; 10; 1; y)
95     ENDSWITCH
96     RETURN

97 LABEL(Thumb)
98     y := MsgLoop[7]
99     y := y & 65535
100    CALL(TypeFruit)
101    RETURN

102 LABEL(LineUp)
103    IF(NOT(y = 1))
104        y := y - 1
105        CALL(TypeFruit)
106    ELSE
107        BEEP
108    ENDIF
109    RETURN

110 LABEL(LineDown)
111    IF(NOT(y = 10))
112        y := y + 1
113        CALL(TypeFruit)
114    ELSE
115        BEEP
116    ENDIF
117    RETURN

118 LABEL(PageUp)
119     y := 1
120     CALL(TypeFruit)
121     RETURN

122 LABEL(PageDown)
123     y := 10
124     CALL(TypeFruit)
125     RETURN

126 LABEL(TypeFruit)
127     vBlank := ""
128     DIALOGADDTTEXT(1000; 0; 60; vOldPos; 40; 10; 1; vBlank)
129     DIALOGADDTTEXT(1000; 0; 60; y * 10; 40; 10; 1; List[y])
130     vOldPos := y * 10
131     RETURN

132 LABEL(MsgBox)

```

```

133 MESSAGEBOX(vStatus; "Message Box"; "Do you want to cancel this message?"; YesNo; I
    IconQuestion!)
134 IF(vStatus = 6)
135     WM_ACTIVATE := 0
136 ENDIF
137 RETURN

```

Line	Description
1-3	Comments about the macro.
2	Macros are generally saved to the default macros directory.
4	WordPerfect identified as the default application.
5-7	Comments identify the opening CALL statements which outline and define the macro's basic operations.
8	Call LABEL(InitializeVariables), which initializes all variables (lines 16-38). A run-time error occurs if you reference a variable that has not been assigned a value. See <i>VARERRCHK</i> in <i>Programming Commands Reference</i> .
9	Call LABEL(CreateDlg) after LABEL(InitializeVariables) returns. LABEL(CreateDlg) creates a dialog box in memory (lines 39-47). It does not display the dialog box.
10	Call LABEL(DisplayDlg) after LABEL(CreateDlg) returns. LABEL(DisplayDlg) displays the dialog box and identifies a callback function. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> .
11	Call LABEL(MessageLoop) after LABEL(DisplayDlg) returns. LABEL(MessageLoop) starts a message loop that suspends the dialog box for callback function (lines 61-74).
12	Call LABEL(QuitMacro) after LABEL(MessageLoop) returns. LABEL(QuitMacro) removes the dialog box from memory and ends the macro.
13-15	Comments identify LABELs that correspond to the opening CALL statements (lines 8-12).
16-38	LABEL(InitializeVariables) initializes all variables.
17-25	Initialize variables with numbers that correspond to positions on the scroll bar. See <i>DIALOGADDSCROLLBAR</i> in <i>Programming Commands Reference</i> .
26	The position of the most recent list item displayed in the dialog box. The position is cleared of text when a new list item is displayed (lines 127-128).
27	The current position for displaying a new list item (line 129).

Line	Description
28-30	Initialize variables to contain Windows message numbers. Using variables makes the macro easier to read. The variable names are user-defined, but match the Windows message names. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> .
31	Initialize the loop variable. In this example, vLoop does not receive a new value. The loop ends when the macro ends (line 57). It is possible to end the message loop and not the macro by changing the value of vLoop.
32	Creates a 10-element array named List.
33-37	Fills array with the names of 10 fruit. Variable vElement identifies the element to fill.
38	RETURN sends macro execution to CALL(CreateDlg).
39-47	LABEL(CreateDlg) creates a dialog box in memory.
42-44	A FORNEXT statement Creates a list of names. The <i>Control</i> parameter of DIALOGADDDTEXT is set to 0 to allow this command to execute in a FORNEXT statement. This is possible for controls that do not accept input.
43	List is a 10-element array. The initial value of variable yPos is also 10. Dividing yPos by 10 reduces yPos to a number that corresponds to an element in the array. DIALOGADDDTEXT uses the number to display the contents of the corresponding element.
44	ENDFOR closes the FORNEXT statement (line 42).
45	Displays the first list item in a second column.
46	Displays the initial thumb position.
47	RETURN sends macro execution to CALL(DisplayDlg).
48-50	LABEL(DisplayDlg) displays the dialog box and identifies the callback function MsgLoop. RETURN sends macro execution to CALL(StartMessageLoop).
51-54	LABEL(StartMessageLoop) starts the message loop that suspends the dialog box for callback functions (lines 61-74). RETURN sends macro execution to CALL(QuitMacro).
55-57	LABEL(QuitMacro) removes the dialog box from memory and ends the macro.
58-60	Comments identify the callback function.
61-74	Callback function MsgLoop responds to three Windows messages, and one <i>Control</i> parameter message. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> .
62	SWITCH tests the fifth element of the callback function array. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> .

Line	Description
63-64	A WM_SYSCOMMAND message is received when the user chooses Close from the system menu box, double-clicks the system menu box, or presses Alt+F4. Macro execution jumps to LABEL(QuitMacro).
65-66	A WM_VSCROLL message is received when the user clicks the scroll bar. The macro calls LABEL(ScrollCtrl).
67-68	A WM_ACTIVATE message is received when the dialog box is activated. The macro calls LABEL(MsgBox).
69	ENDSWITCH closes the SWITCH statement (line 62).
70-73	SWITCH tests the third element of the callback function array, and determines when Cancel is chosen. See <i>DIALOGDISPLAY</i> in <i>Programming Commands Reference</i> . An IF statement also works here. There is often more than one choice to test.
74	RETURN sends macro execution to the message loop (lines 52-53).
75-96	LABEL(ScrollCtrl) tests the <i>wParam</i> of the MW_VSCROLL message (line 76).
76	SWITCH tests the sixth element (<i>wParam</i> value passed with WM_VSCROLL) of the callback function array, to determine mouse position when clicked.
77-94	See <i>DIALOGADDSCROLLBAR</i> in <i>Programming Commands Reference</i> for details about WM_VSCROLL messages.
77-78	Scroll bar's right arrow clicked. Call LABEL(LineUp).
79-80	Scroll bar's left arrow clicked. Call LABEL(LineDown).
81-82	Area between the right arrow and thumb clicked. Call LABEL(PageUp).
83-84	Area between the left arrow and thumb clicked. Call LABEL(PageDown).
85-86	Position of thumb after it is dragged, passed as the low-order word of <i>lParam</i> in the seventh element of the callback function array. Call LABEL(Thumb).
87-88	Current position of thumb as it is dragged, passed as the low-order word of <i>lParam</i> in the 7th element of the callback function array. Call LABEL(Thumb). Lines 87-88 call the same function as lines 85-86, and have comment marks (<i>//</i>). To test lines 87-88, add comment marks (<i>//</i>) to lines 85-86 and remove them from 87 and 88 (see <i>// (Comment)</i> in <i>Programming Commands Reference</i>).
89-90	Home key pressed while the scroll bar has the input focus. Call LABEL(PageUp).
91-92	End key pressed while the scroll bar has the input focus. Call LABEL(PageDown).

Line	Description
93-94	Scroll bar activity has ended. Display the current thumb position, contained in variable <code>y</code> .
95	ENDSWITCH closes the SWITCH statement (line 76).
96	RETURN sends macro execution to LABEL(MsgLoop) callback function.
97-101	LABEL(Thumb) retrieves the thumb position after it is dragged (MsgLoop[6] = SB_THUMBPOSITION), or as it is dragged (MsgLoop[6] = SB_THUMBTRACK).
98	Assign the value of MsgLoop[7] (element 7 of the callback function array) to variable <code>y</code> .
99	Calculate the low-order word of variable <code>y</code> using the bitwise AND operator. See <i>Bitwise Operators</i> in <i>Variables, Constants, Operators, and Expressions</i> . The value assigned to <code>y</code> is the current thumb position.
100	Call LABEL(TypeFruit) to display the list item that corresponds to the current thumb position.
101	RETURN sends macro execution to LABEL(ScrollCtrl).
102-109	LABEL(LineUp) is called when the right arrow is clicked, to make the current line (value of <code>y</code>) equal to the current thumb position.
103-105	If the current line is not 1, assign the value of <code>y - 1</code> to <code>y</code> , and call LABEL(TypeFruit) to display the list item that corresponds to the current thumb position.
106-107	Beep if the value of <code>y</code> equals 1 (if the first item is displayed). (Try changing this to display the last item.)
108	ENDIF closes the IF statement (line 103).
109	RETURN sends macro execution to LABEL(ScrollCtrl).
110-117	LABEL(LineDown) is called when the left arrow is clicked, to make the current line (value of <code>y</code>) equal to the current thumb position.
111-113	If the current line is not 10, assign the value of <code>y + 1</code> to <code>y</code> , and call LABEL(TypeFruit) to display the list item that corresponds to the current thumb position.
114-115	Beep if the value of <code>y</code> equals 10 (if the last item is displayed). (Try changing this to display the first item.)
116	ENDIF closes the IF statement (line 111).
117	RETURN sends macro execution to LABEL(ScrollCtrl).
118-121	LABEL(PageUp) is called when the area between the right arrow and thumb is clicked.
119	Assign 1 to variable <code>y</code> to make 1 the current line and thumb position.
120	Call LABEL(TypeFruit) to display the first list item on the first line.

Line	Description
121	RETURN sends macro execution to LABEL(ScrollCtrl).
122-125	LABEL(PageDown) is called when the area between the left arrow and thumb is clicked.
123	Assign 10 to variable y to make 10 the current line and thumb position.
124	Call LABEL(TypeFruit) to display the last list item on the last line.
125	RETURN sends macro execution to LABEL(ScrollCtrl).
126-131	LABEL(TypeFruit) displays a list item on the line that corresponds to the current thumb position.
127	Assign blank spaces to variable vBlank, to erase the previous list item when a new one is displayed.
128	Erase the previous list item. The vertical position is contained in variable vOldPos, that has an initial value of 10 (line 26). This value is updated (line 130) every time LABEL(TypeFruit) is called.
129	Type a new list item at the current thumb position times 10 (the current line position). The current thumb position (value of y) identifies the current list item (List[y]).
130	Assign the current line position to variable vOldPos, to be used (line 128) when this subroutine is called again.
131	RETURN sends macro execution to the subroutine's caller.
132-137	LABEL(MsgBox) is called when the dialog box is activated (lines 67-68). If you choose No, the dialog box is activated when the message box is dismissed and LABEL(MsgBox) is called again. The message box is called until the user chooses Yes (lines 134-136).
133	See <i>MESSAGEBOX</i> in <i>Programming Commands Reference</i> .
134-136	If variable vStatus equals 6 (Yes chosen), assign 0 to WM_ACTIVATE. Changing the value of WM_ACTIVATE invalidates it as a Windows activate message (lines 30 and 67-68).
137	RETURN sends macro execution to LABEL(MsgLoop) callback function.

Dynamic Data Exchange

DDE is a method of the Windows operating system that transfers commands and other information between applications. For example, you can use DDE to execute WP Office commands from WordPerfect.

CALENDAR.WCM

The following example opens WP Office for Windows 4.0a Day calendar. Line-by-line comments follow the macro text. If you want to play CALENDAR.WCM, copy it from the WordPerfect for Windows 6.0a Online Macros Manual. See *Programming Commands Reference* for detailed command descriptions.

```
1 // Name: CALENDAR.WCM
2 // Directory: C:\WPWIN60\MACROS
3 // Description: Open WP Office 4.0a Day calendar
4 // Method: Dynamic Data Exchange

5 APPLICATION(WP; "WordPerfect"; Default; "US")

6 //*****
7 // MAIN CALL STATEMENTS
8 //*****

9 CALL(IsOfficeRunning)
10 CALL(OpenDayCalendar)
11 CALL(QuitMacro)

12 //*****
13 // MAIN LABEL STATEMENTS
14 //*****

15 LABEL(IsOfficeRunning)
16 IF(APPLICATE("WordPerfect Office 4.0a") = 0)
17   vPath := "c:\office40\"
18   vResult := APPEXECUTEEXT(vPath + "ofwin.exe"; 0)
19   IF(vResult < 32)
20     SWITCH(vResult)
21       CASEOF 2: CALL(FileNotFound)
22       CASEOF 3: CALL(PathNotFound)
23       DEFAULT: CALL(ErrorMsg)
24   ENDSWITCH
25   ENDIF
26   ENDIF
27   RETURN

28 LABEL(OpenDayCalendar)
29 ASSIGN(hDay; APPLICATE("Day"))
30 IF(hDay)
31   APPACTIVATE(hDay)
32 ELSE
33   ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "COMMAND"))
34   IF(hConv)
35     DDEEXECUTE(hConv; "ViewOpenNamed(ViewName: ""Day""; ViewType:Calendar)")
36   ELSE
37     CALL(QuitMacro)
38   ENDIF
39 ENDIF
40 RETURN

41 LABEL(QuitMacro)
```



```

42 DDETERMINATEALL
43 QUIT

44 //*****
45 // SECONDARY LABEL STATEMENTS
46 //*****

47 LABEL(FileNotFound)
48 BEEP
49 ASSIGN(vMessage; "File not found - ending macro")
50 GO(DisplayMessage)

51 LABEL(PathNotFound)
52 BEEP
53 ASSIGN(vMessage; "Path not found - ending macro")
54 GO(DisplayMessage)

55 LABEL(ErrorMessage)
56 BEEP
57 ASSIGN(vMessage; "Unable to launch WP Office 4.0a - ending macro")
58 GO(DisplayMessage)

59 LABEL(DisplayMessage)
60 MESSAGEBOX(var; "ERROR"; vMessage; Ok! | IconExclamation!)
61 GO(QuitMacro)

```

Line Description

- 1-4 Comments about the macro.
- 2 Macros are generally saved to the default macros directory.
- 5 WordPerfect identified as the default application.
- 6-8 Comments identify the opening CALL statements, which outline and define the macro's basic operations.
- 9 Call LABEL(IsOfficeRunning) to determine if Office is running.
- 10 Call LABEL(OpenDayCalendar) to initiate a DDE conversation with Office and send an Office command to open the Day calendar view.
- 11 Call LABEL(QuitMacro) to end the macro.
- 12-14 Comments identify the main LABEL statements.
- 15-27 LABEL(IsOfficeRunning) determines if Office is running, and, if not, starts it.
- 16 APPLOCATE returns the Office 4.0a window handle if the window title is found, or 0 if not. If 0 is returned, the next statement is executed (line 17). If a window handle is returned, RETURN (line 27) sends macro execution to CALL(OpenDayCalendar). Another way to write the statement is:

```

vResult := APPLOCATE("WordPerfect Office 4.0a")
IF(vResult = 0)

```

Line	Description
17	If necessary, change this line to the directory that contains the WP Office program files.
18	APPEXECUTEEXT returns the application handle if Office is started (line 16), or an error value less than 32 if not. See <i>APPEXECUTEEXT</i> in <i>Programming Commands Reference</i> .
19	If vResult is less than 32 (error occurred), execute SWITCH statement (lines 20-24).
20-23	IF vResult equals 2, call LABEL(FileNotFound). If vResult equals 3, call LABEL(PathNotFound). For any value less than 32, call LABEL(ErrorMessage). See <i>APPEXECUTEEXT</i> in <i>Programming Commands Reference</i> for a list of error values.
24	ENDSWITCH closes the SWITCH statement (line 20).
25	ENDIF closes the IF statement (line 19).
26	ENDIF closes the IF statement (line 16).
27	RETURN sends macro execution to CALL(OpenDayCalendar).
28-40	LABEL(OpenDayCalendar) determines if the Day calendar view is already open.
29	APPLOCATE returns the Calendar view window handle if the window title is found, or 0 if not. The handle or 0 is assigned to variable hDay.
30-31	If variable hDay equals a window handle (a non-zero number), use APPACTIVATE to activate the Calendar view. The statement IF(hDay) is equivalent to IF(hDay <> 0).
32-33	IF variable hDay equals 0 (ELSE), initiate a conversation with Office and assign the result to variable hConv.
34-35	If variable hConv equals a window handle (a non-zero number), execute the ViewOpenedNamed Office command.
36-37	If variable hConv equals 0 (conversation not initiated), call LABEL(QuitMacro).
38	ENDIF closes IF statement (line 34).
39	ENDIF closes IF statement (line 30).
40	RETURN sends macro execution to CALL(QuitMacro).
41-43	LABEL(QuitMacro) ends the macro.
44-46	Comments identify the SECONDARY label statements.
47-50	LABEL(FileNotFound) beeps, assigns a file not found message to variable vMessage, and sends macro execution to LABEL(DisplayMessage).
51-54	LABEL(PathNotFound) beeps, assigns a path not found message to variable vMessage, and sends macro execution to LABEL(DisplayMessage).

Line	Description
55-58	LABEL(ErrorMessage) beeps, assigns an unknown error message to variable vMessage, and sends macro execution to LABEL(DisplayMessage).
59-61	LABEL(DisplayMessage) displays the appropriate error message, and sends macro execution to LABEL(QuitMacro).
60	MESSAGEBOX displays the contents of variable vMessage (lines 49, 53, and 57).

Assigning a Macro

Keystroke Macros

Macros assigned to a key on a keyboard.

To assign a macro to a key:

- 1 Choose **File, Preferences**, then double-click **Keyboard**.
- 2 Choose **Create**, enter a keyboard definition name.
or
Select a keyboard definition, then choose **Edit**.
- 3 Select a key, select **Play a Macro**, choose **Assign Macro**, enter the path and filename of a macro you want to play, then choose **Select**.

To play a keystroke macro, press the key the macro is assigned to.

Button Bar Macros

Macros assigned to a Button Bar.

To assign a macro to a Button Bar:

- 1 Choose **File, Preferences**, then double-click the **Button Bar** icon.
- 2 Choose **Create**, enter a Button Bar name.
or
Select a Button Bar definition, then choose **Edit**.
- 3 Select **Play a Macro**, choose **Add Macro**, enter the path and filename of a macro you want to add, then choose **Select**.

To play a Button Bar macro, choose the button the macro is assigned to.

Menu Bar Macros

Macros assigned to the macro menu or to a customized menu bar. You can assign up to nine macros to the macro menu.

Macro Menu:

- 1 Choose **Tools, Macro**, then **Menu**.
- 2 Select a position on the list where you want to insert the macro, then choose **Insert**.

or

Choose **Insert** if no macros are listed.

- 3** Enter the path and filename of a macro you want to add, then choose **Select**.

To play a menu macro, select the menu item the macro is assigned to.

Customized Menu Bar:

- 1** Choose **File, Preferences**, then double-click the **Menu Bar** icon.

- 2** Choose **Create**, then enter a menu bar name.

or

Select a menu bar layout, then choose **Edit**.

- 3** To add a cascading menu, drag the menu icon to the menu bar.

or

Skip to step 5.

- 4** Double-click the new menu, enter the name of the menu item, then enter a prompt message.
- 5** Select **Play a Macro**, choose **Assign Macro**, enter a macro path and filename, then choose **Select**.
- 6** Move the macro name to the new menu (step 3), or to another menu by dragging it.
- 7** Double-click the macro name, enter the name of the menu item, then enter a prompt message.
- 8** Choose **OK**, highlight a menu bar layout, choose **Select**, then **Close**.

Programming Commands Reference

.....

// (Comment)

Purpose	The compiler ignores all text between // and the next hard return [HRt].
Syntax	// ... <i>comments</i> ... [HRt]
Description	Comment text does not affect macro execution. The comment line, including spaces, has a maximum length of 512 characters.
Example	See the Example heading for other programming commands.

APPACTIVATE

Purpose	Activates and displays a window.
Syntax	APPACTIVATE (<i>WindowHandle</i>)
Description	Use APPACTIVATE to return the window handle in a variable. <pre> ASSIGN(hWnd; APPLOCATE("Program Manager")) IF(hWnd) // if hWnd not equal to 0 APPACTIVATE(hWnd) ELSE QUII ENDIF</pre>
Parameters	
<i>WindowHandle</i>	Numeric Expression: The window handle of the window to activate (see APPLOCATE).

Example	<pre>APPLICATION(A1; "WordPerfect"; Default; "US") // Execute Windows Calculator // Locate and activate 3 times // Demonstrate APPACTIVATE command ASSIGN(hWP; APPLOCATE("")) APPEXECUTE("c:\windows\calc.exe") ASSIGN(hCalc; APPLOCATE("Calculator")) FORNEXT(vCtrl; 1; 2; 1) WAIT(10) APPACTIVATE(hWP) WAIT(10) APPACTIVATE(hCalc) ENDFOR</pre>
----------------	--

See Also

♦ APPEXECUTE ♦ APPLOCATE

APPEXECUTE

Purpose Starts an application and displays it.

Syntax APPEXECUTE (*CommandLine*)

Description The macro terminates if the application is not successfully started. APPEXECUTE is the equivalent of the Windows Program Manager Run command.

Parameters

CommandLine Character Expression: The path and name of a program file.

Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Execute Windows Calculator
// Locate and activate 3 times
// Demonstrate APPEXECUTE command

ASSIGN(hWP; APPLOCATE("**"))
APPEXECUTE("c:\windows\calc.exe")
ASSIGN(hCalc; APPLOCATE("Calculator"))
FORNEXT(vCtrl; 1; 2; 1)
    WAIT(10)
    APPACTIVATE(hWP)
    WAIT(10)
    APPACTIVATE(hCalc)
ENDFOR
```

See Also

♦ APPACTIVATE ♦ APPLOCATE ♦ APPEXECUTEEXT

APPEXECUTEEXT

Purpose Starts an application and specifies how its window is displayed.

Syntax ReturnValue := APPEXECUTEEXT (*CommandLine; ShowWindow*)

Returns The application handle (unique number) if the application is successfully started, or an error value less than 32 if not. For the list of error values, see Windows 3.1 WinExec documentation (Programmer's Reference, Volume 2: Functions). A partial list follows:

- 0 System out of memory or program file is corrupt.
- 2 File not found.
- 3 Path not found.
- 8 Insufficient memory to start application.
- 10 Incorrect Windows version.
- 11 Non-Windows application.

Check error values with code similar to the following:

```
vError := APPEXECUTEEXT(CommandLine; ShowWindow)
SWITCH(vError)
  CASEOF 0: ...statement block...
  CASEOF 2: ...statement block...
ENDSWITCH
```

Parameters

CommandLine

Character Expression: The path and name of a program file.

ShowWindow

Numeric Expression: Specifies how an application's window is displayed. The values are,

- 0 Hide window. APPLOCATE and DDEINITIATE return the window handle of a hidden window. DDEEXECUTE sends a command string to a hidden window. APPACTIVATE activates a hidden window. Accelerator keys choose a hidden window's menu options.
- 1 Show window at normal size.
- 2 Minimize window to an icon.
- 3 Maximize window to full screen size.

Example

```
APPLICATION (A1; "WordPerfect"; Default: "US")
// Execute Windows CARDFILE.EXE three times: full screen, normal size, and to an icon
// FORNEXT decrements initial value
// Demonstrate APPEXECUTEEXT command

FORNEXT(vCtrl; 3; 1; -1)
  ASSIGN(vReturn; APPEXECUTEEXT("c:\windows\cardfile.exe"; vCtrl))
  IF(vReturn < 32)
    BEEP
    MESSAGEBOX(vStatus; "ERROR"; "Unable to execute Cardfile.exe"; IconExclamation!)
  QUIT
ENDIF
ENDFOR
```

See Also

♦ APPACTIVATE ♦ APPLOCATE ♦ APPEXECUTE ♦ DDEEXECUTE

APPLICATION

Purpose

Identifies, for the compiler, an application to use in a macro.

Syntax

APPLICATION (*ProductPrefix*; *ApplicationName*; Default!; *Language*)

Description

APPLICATION is a non-executable statement that can occur anywhere in a macro, but must precede product commands to the application it identifies.

Applications used in the macro automatically start unless already active. The user cannot terminate an application used in a macro until the macro ends.

Parameters*ProductPrefix*

Character Expression: A two-character identifier that begins with a letter. The second character can be a letter or number. *ProductPrefix* directs product commands to non-default applications. It is also used in NEWDEFAULT and ENDAPP statements.

Use a period to attach *ProductPrefix* to a product command. If WordPerfect is the application and WP the product prefix, WP.AboutDlg is a valid command statement. A recorded macro returns a product prefix of A1, A2, A3, and so forth, depending on the number of applications involved.

ApplicationName

Character Expression: The name of an application. For example, "WordPerfect" for WordPerfect 6.0 for Windows.

Default!

An enumeration that identifies a default application. Product commands to the default application do not require a product prefix. There can be only one default application.

Language

Character Expression: The application's language code. For example, "US" is the language code for United States English.

Example

```
APPLICATION (A1; "WordPerfect"; Default; "US")
// Execute Windows Cardfile three times: full screen, normal size, and to an icon
// FORNEXT decrements initial value
// Demonstrate APPLICATION command

FORNEXT(vCtrl; 3; 1; -1)
  ASSIGN(vReturn; APPEXECUTEext("c:\windows\cardfile.exe"; vCtrl))
  IF(vReturn < 32)
    BEEP
    MESSAGEBOX(vStatus; "ERROR"; "Unable to execute Cardfile.exe"; IconExclamation!)
  QUIT
ENDIF
ENDFOR
```

See Also

♦ ENDAPP ♦ NEWDEFAULT

APPLOCATE

Purpose	Returns a window handle.
Syntax	ReturnValue := APPLOCATE (<i>WindowTitle</i>)
Returns	The window handle of the window found in the search, or 0 if no window is found.
Description	<p>APPLOCATE compares a name to the title bar text of all open windows. If a match is found, the window handle of the matching window is returned. If the search criteria matches more than one window, the window handle of the most recently opened window is returned. If no match is found, zero is returned.</p> <p>You can use an asterisk (*) as a wildcard character at any position in the APPLOCATE parameter. If the parameter contains only an asterisk, APPLOCATE returns the window handle of the active window.</p>

Parameters

<i>WindowTitle</i>	Character Expression: The window title to find. The match between this parameter and the title must be exact unless you use an asterisk as a wildcard character. Titles are case sensitive.
--------------------	---

Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Execute Windows Calculator
// Locate and activate 3 times
// Demonstrate APPLOCATE command

ASSIGN(hWP; APPLOCATE("**"))
APPEXECUTE("c:\windows\calc.exe")
ASSIGN(hCalc; APPLOCATE("Calculator"))
FORNEXT(vCtrl; 1; 2; 1)
    WAIT(10)
    APPACTIVATE(hWP)
    WAIT(10)
    APPACTIVATE(hCalc)
ENDFOR
```

See Also ♦ APPACTIVATE ♦ APPEXECUTE

ASSERT

Purpose	Creates a Cancel, Error, or Not Found condition.
----------------	--

SyntaxASSERT (*Condition*)

Description

A Cancel, Error, or Not Found condition stops a macro unless preceded by ONCANCEL, ONERROR, or ONNOTFOUND, which direct macro execution to a specified LABEL. ASSERT has no effect when preceded by CANCEL(Off!), ERROR(Off!), or NOTFOUND(Off!).

An error value is assigned to variable ErrorNumber when one of the three conditions occurs. The values are,

- 1 Cancel condition.
- 2 Error condition.
- 7 Not found condition.

You cannot assign a value to ErrorNumber.

Use code similar to the following to check all three conditions with one subroutine (see *Subroutines in Macro Control Statements*, and line 44 of ASSERT.WCM example in *Making Decisions with MENU*).

```
ONERROR(Condition)
ONCANCEL(Condition)
ONNOTFOUND(Condition)

...other statements...

LABEL(Condition)
SWITCH(ErrorNumber)
  CASEOF 1: ...statement block...
  CASEOF 2: ...statement block...
  CASEOF 7: ...statement block...
ENDSWITCH
```

Parameters*Condition*

Enumeration: A condition to assert. The enumerations are,

CancelCondition!	Stops a macro unless preceded by ONCANCEL.
ErrorCondition!	Stops a macro unless preceded by ONERROR.
NotFoundCondition!	Stops a macro unless preceded by ONNOTFOUND.

Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate ASSERT command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
  MENU(vPick; Digit;;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
  SWITCH(vPick)
    CASEOF 1: ASSERT(CancelCondition!)
```

```

CASEOF 2: ASSERT(ErrorCondition!)
CASEOF 3: ASSERT(NotFoundCondition!)
CASEOF 4: QUIT
DEFAULT: GO(Condition)
ENDSWITCH

```

```

LABEL(Condition)
SWITCH(ErrorNumber)
CASEOF 1: vMsg := "1 (Cancel condition)"
CASEOF 2: vMsg := "2 (Error condition)"
CASEOF 7: vMsg := "3 (Not Found condition)"
DEFAULT: vMsg := "?"
ENDSWITCH
MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
GO(Start)

```

See Also

♦ CANCEL ♦ ERROR ♦ LABEL ♦ NOTFOUND ♦ ONCANCEL
 ♦ ONERROR ♦ ONNOTFOUND ♦ RETURN ♦ SWITCH

ASSIGN

Purpose

Assigns a value to a variable.

Syntax

ASSIGN (*VariableName*; *Value*)

Description

ASSIGN is equivalent to the assignment operator (:=). Consider the following four statements and their equivalents:

```

ASSIGN(A; B) // assign the contents of B to A
ASSIGN(A; 487) // assign 487 to A
ASSIGN(A; B + 487) // assign the contents of B + 487 to A
ASSIGN(A; "John Doe") // assign the name John Doe to A

```

```

A := B // assign the contents of B to A
A := 487 // assign 487 to A
A := B + 487 // assign the contents of B + 487 to A
A := "John Doe" // assign the name John Doe to A

```

Parameters

VariableName

Variable: Contains the contents of *Value*.

Value

Any: A variable, constant, character string, or expression. *Value* has a maximum length of 512 characters.

Example

```
APPLICATION (A1; "WordPerfect"; Default; "US")
// Execute Windows Cardfile three times: full screen, normal size, and to an icon
// FORNEXT decrements initial value
// Demonstrate ASSIGN command

FORNEXT(vCtrl; 3; 1; -1)
  ASSIGN(vReturn; APPEXECUTEEXT("c:\windows\cardfile.exe"; vCtrl))
  IF(vReturn < 32)
    BEEP
    MESSAGEBOX(vStatus; "ERROR"; "Unable to execute Cardfile.exe"; IconExclamation!)
  QUIT
ENDIF
ENDFOR
```

See Also

♦ DISCARD ♦ GLOBAL ♦ INDIRECT ♦ LOCAL ♦ PERSIST ♦ PERSISTALL

BEEP

Purpose

Causes a computer speaker or installed sound board to beep.

Syntax

BEEP

Description

BEEP is commonly used with warning messages, or to signal when a macro pauses or finishes a task.

Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Beep vCount times - BREAK when vCount equals 4
// Remember: functions do not execute unless called
// Demonstrate BEEP command

FUNCTION BeepBeep(vCount)
  REPEAT
    BEEP
    WAIT(3)
    vCount := vCount - 1
  UNTIL(vCount = 0)
  RETURN
ENDFUNC

FOREACH(vCount; {1; 2; 3; 4; 5})
  IF(vCount = 4)
    BREAK
  ENDIF
  BeepBeep(vCount)
  WAIT(5)
ENDFOR
```

```
MESSAGEBOX(x; "BREAK"; "Variable vCount equals 4"; IconInformation!)  
QUIT
```

BIFFILEPATH

Purpose	Retrieves the path and filename of a Binary Initialization File (BIF).
Syntax	BIFFILEPATH (<i>BIFFFilePath</i> ; <i>BIFFFileType</i>)
Description	See <i>WordPerfect 6.0 for Windows Software Developer's Kit</i> for more information about the BIF approach to initialization files.
Parameters	
<i>BIFFFilePath</i>	Character Expression: The BIF path and filename is returned in this variable.
<i>BIFFFileType</i>	Enumeration: The BIF type that is returned to <i>BIFFFilePath</i> . A user can have only two active BIF files. The enumerations are, Private! A BIF file containing settings that apply to a single user. Public! A BIF file containing settings that are shared by many, possibly concurrent, users.
Example	<pre>APPLICATION(A1; "WordPerfect"; Default; "US") // Demonstrate BIFFILEPATH command HdRtrn := NTOC(0F90Ah) BIFFILEPATH(vPrivate; Private!) BIFFILEPATH(vPublic; Public!) MESSAGEBOX(x; "BIFFILEPATH COMMAND"; "Private BIF path and filename: " + vPrivate + HdRtrn + HdRtrn + "Public BIF path and filename: " + vPublic; IconInformation!)</pre>
See Also	♦ BIFINFO ♦ BIFREAD ♦ BIFWRITE

BIFINFO

Purpose	Retrieves information from a Binary Initialization File (BIF).
Syntax	BIFINFO (<i>Status</i> ; <i>Group</i> ; <i>Section</i> ; <i>Item</i> ; <i>DateTime</i> ; <i>ItemFlags</i> ; <i>ItemLength</i> ; <i>ItemType</i> ; <i>UseCount</i> ; <i>ItemInfoType</i> ; <i>BIFFile</i>).

Description See *WordPerfect 6.0 for Windows Software Developer's Kit* for more information about the BIF approach to initialization files.

Parameters

Status Numeric Expression: The status of a BIF call is returned in this variable. The values are,

- 0 Call was successful.
- 2 Length of Group, Section, or Item exceeds buffer size.
- 3 Requested Group, Section, or Item not found.
- 4 Unknown error.

Group Character Expression: The name of a Group. If this parameter is not used, the number of Group names is returned to the *UseCount* parameter.

Section Character Expression: The name of a Section. If this parameter is not used, the number of Section names in the specified Group is returned to the *UseCount* parameter.

Item Character Expression: The name of an Item. If this parameter is not used, the number of Item names in the specified Group and Section is returned to the *UseCount* parameter.

DateTime Variable: The date and time the item was created is returned in this variable. If Group, Section, and Item are not used, the current date and time is returned.

ItemFlags Variable: The flags associated with an item are returned in this variable. If Group, Section, and Item are not used, 0 is returned.

ItemLength Variable: The length of an item entry is returned in this variable. If Group, Section, and Item are not used, 0 is returned.

ItemType Variable: The item type is returned in this variable. If Group, Section, and Item are not used, 0 is returned.

UseCount Variable: If Group, Section, or Item is not used, the number of Groups, Sections, or Items in a BIF is returned in this variable (see *Group*, *Section*, and *Item* parameters). Otherwise, 0 is returned.

ItemInfoType Enumeration: The standard BIF types to examine. The enumerations and descriptions are,

- Both! Look in both standard BIFs. Only valid when counting Groups, Sections, or Items.
- Private! Look in Private BIF only.
- Public! Look in Public BIF only.

BIFFile Character Expression: The name of a non-standard BIF.

Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")  
// This example uses the BIF created with the BIFWRITE example  
// Demonstrate BIFINFO command
```



```
HdRtrn := NTOC(0F90Ah)
```

```
BIFINFO(vStatus; "Group 1"; "Section 1"; "Item 1"; vDate; vFlags; vLength; vType; vCount;;  
"c:\wpwin60\macros\test.bif")
```

```
MESSAGEBOX(x; "BIFINFO COMMAND - Test.bif file"; "Date/time created: " + vDate + HdRtrn +  
HdRtrn + "Flags: " + vFlags + HdRtrn + HdRtrn + "Length of item: " + vLength + HdRtrn + HdRtrn +  
"Item type: " + vType + HdRtrn + HdRtrn + "Count: " + vCount + HdRtrn)
```

See Also

♦ BIFFILEPATH ♦ BIFREAD ♦ BIFWRITE

BIFREAD

Purpose Reads a Binary Initialization File (BIF), and returns Group, Section, or Item names, or item value.

Syntax BIFREAD (*Status*; *Group*; *Section*; *Item*; *Value*; *ItemType*; *NameSeparator*;
NameListType; *BIFFile*)

Description Group, the highest BIF level. For example, in the standard BIF, Group is associated with a WordPerfect product such as WP Macros, WP Shared Code, and WordPerfect.

Section, the next BIF level after Group, is associated with Group subdivisions. For example, WordPerfect Sections include features such as Ruler, Power Bar, Table of Authorities, Repeat Value, and Files.

Item, the lowest BIF level, is associated with a Section entry. For example, in the WordPerfect Section named Files, Items include Hyphenation Directory, Timed Document Backup, Macros Directory, and Template File.

Item levels contain values. For example, the value of Macros Directory may be C:\WPWIN60\MACROS.

Group, Section, and Item names are subject to change. See an application's documentation for more information.

Parameters

Status Variable: The status of a BIF call is returned in this variable. The values are,

- 0 (or positive integer) Number of bytes read from the BIF.
- 1 *ItemType* does not match returned item type (see *ItemType* parameter).
- 2 Length of Group, Section, or Item exceeds buffer size.
- 3 Requested Group, Section, or Item was not found.
- 4 Unknown error.

<i>Group</i>	Character Expression: Contains a Group name. If <i>Group</i> is not used, the BIF Group names are returned in a variable (see <i>Value</i> parameter).
<i>Section</i>	Character Expression: Contains a Section name. If <i>Group</i> is used and <i>Section</i> is not, the BIF Section names for the specified Group are returned in a variable (see <i>Value</i> parameter).
<i>Item</i>	Character Expression: Contains an Item name. If <i>Group</i> and <i>Section</i> are used and <i>Item</i> is not, the BIF Item names for the specified Group and Section are returned in a variable (see <i>Value</i> parameter).
<i>Value</i>	Variable: The <i>Group</i> , <i>Section</i> , or <i>Item</i> names are returned in this variable, depending on which one is not used. If <i>Group</i> , <i>Section</i> , and <i>Item</i> are used, the <i>Item</i> value is returned. For example, given that "c:\wpwin60\macros\" is the default macros directory in a standard BIF, <pre style="margin-left: 40px;">BIFREAD(vStatus; "WordPerfect"; "Files"; "Macro Directory"; vEntry; AnsiString!;;)</pre> returns 18 to vStatus (the number of bytes returned in <i>Value</i>) and "C:\WPWIN60\MACROS\" to the <i>Value</i> parameter, vEntry.
<i>ItemType</i>	Variable: The expected item type to return in a variable (see <i>Value</i> parameter). <i>ItemType</i> is not used if Group, Section, or Item is not used. The values and enumerations are, <pre style="margin-left: 40px;">0 Any! 1 Boolean! 2 SignedByte! 3 UnsignedByte! 4 SignedWord! 5 UnsignedWord! 6 SignedDWord! 7 UnsignedDWord! 8 Float4! 9 Float8! 10 AnsiString! 11 WPWordString! 12 BagOfbits!</pre>
<i>NameSeparator</i>	Character Expression: A delimiter string that separates Group, Section, or Item names. The delimiter is not limited to a single character. If not used, the default is a semicolon. <i>NameSeparator</i> is not used when returning an item value (Group, Section, and Item are used).
<i>NameListType</i>	Enumeration: A standard BIF to read. Standard BIFs are associated with WordPerfect products (see <i>BIFFile</i> parameter). The values are, <pre style="margin-left: 40px;">Both! Look in both standard BIFs. Private! Look in Private BIF only. Public! Look in Public BIF only.</pre>

Character Expression: The name of a non-standard BIF (a user-defined or non-WordPerfect product BIF). If both parameters are used, *BifFile* overrides *NameListType*. If neither parameter is used, the default is *NameListType*.

Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// This example uses the BIF created with the BIFWRITE example
// Demonstrate BIFREAD command

ONCANCEL(QuitMacro)
VARERRCHK(Off)
REPEAT
  GETNUMBER(vNmbr; "Type the number of a group to read:" "BIFREAD COMMAND")
  IF((vNmbr > 5) OR (vNmbr < 1))
    BEEP
    NEXT
  ENDIF
  BIFREAD(vResult; "Group " + vNmbr; "Section " + vNmbr; "Item " + vNmbr; vItemValue;
  WPWordString!;; "c:\wpwin60\macros\test.bif")
  IF(vResult < 0)
    BEEP
    MESSAGEBOX(x; "ERROR"; "Unknown error occurred"; IconExclamation!)
    QUIT
  ELSE
    MESSAGEBOX(x; "Item Value"; "The value of Group, Section, and Item " + vNmbr + " is: " +
    vItemValue; IconInformation! | RetryCancel!)
  ENDIF
UNTIL(x = 2)

LABEL(QuitMacro)
QUIT
```

See Also

• BIFFILEPATH • BIFINFO • BIFWRITE

BIFWRITE**Purpose**

Writes data to a Binary Initialization File (BIF).

IMPORTANT: To write data to a BIF, you must use the Group, Section, Item, and Value parameters. If you do not specify a value for each parameter, you will delete a level or item from the BIF. See parameter descriptions below.

Syntax

BIFWRITE (*Status*, *Group*; *Section*; *Item*; *Value*; *ItemType*; *ItemFlags*, *BIFFile*)

Description

See BIFREAD Comments for a description of BIF Groups, Sections, and Items.

Parameters

- Status* Variable: The status of a BIF call is returned in this variable. The values are,
- 0 (or positive integer) Number of bytes written to the BIF.
 - 1 Unknown Group, Section, or Item type.
 - 2 Requested Group, Section, or Item not found.
 - 3 Unknown error.
- Group* Character Expression: A Group name. To delete a Group, specify the Group name and do not use the Section parameter. To delete all Groups, Sections, and Items, do not use the Group parameter.
- Section* Character Expression: A Section name. The Group name must be used. To delete a Section, specify the Section name and do not use the Item parameter.
- Item* Character Expression: An Item name. The Group and Section names must be used. To delete an Item, specify the Item name and do not use the Value parameter.
- Value* Character Expression: The Item value to write to the BIF. For example,
- ```
BIFWRITE(vStatus; "WordPerfect"; "Files"; "Macro Directory"; "C:\WPWIN60\MACROS";
AnsiString!; 0;)
```
- writes "C:\WPWIN60\MACROS" to a private BIF. It is the value of the "Macro Directory" item in the "Files" section of the "WordPerfect" group.
- IMPORTANT:** Do not write to a standard BIF while the associated WordPerfect application is running. Standard BIFs remain open while Shared Code is running. Play the previous example from Macro Facility.
- ItemType* Enumeration: The *Value* data type to write to the BIF. The values and enumerations are,
- 0 Any!
  - 1 Boolean!
  - 2 SignedByte!
  - 3 UnsignedByte!
  - 4 SignedWord!
  - 5 UnsignedWord!
  - 6 SignedDWord!
  - 7 UnsignedDWord!
  - 8 Float4!
  - 9 Float8!
  - 10 AnsiString!
  - 11 WPWordString!
  - 12 BagOfbits!
- If you write a new value to a BIF item, you must specify the same type as the previous item. If not specified, *ItemType* is determined by the default value for each data type.
- If *ItemType* does not match the current value type, it is converted if possible. If not converted properly, you may not be able to read the value back later (see *BIFFILEREAD*).

## ItemFlags

Enumeration: Flags (attributes) only affect the public BIF. Value 4 is processed when an item is read from a BIF. All other values are processed when Shared Code starts. The values and enumerations are,

- |   |                      |                                                                                                                      |
|---|----------------------|----------------------------------------------------------------------------------------------------------------------|
| 0 | None!                | No flags.                                                                                                            |
| 1 | DeletePrivate!       | Removes an item in a private BIF that has the same name as an item in the public BIF.                                |
| 2 | OverridePrivate!     | The public BIF item has precedence over an item with the same name in a private BIF. Normally, the opposite is true. |
| 3 | DistributeToPrivate! | An item is copied to the private BIF when opened.                                                                    |

## BIFFile

Character Expression: Contains the name of a non-standard BIF to write to. If the BIF does not exist, it is created with specified Group, Section, and Item names, and Item value. To write to a private BIF, do not use this parameter (see BIFREAD, *NameListType* parameter).

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Create a BIF file with five groups, sections, items, and item values
// Demonstrate BIFWRITE command

VARERRCHK(Off!)
HdRtrn := NTOC(0F90Ah)
vFile := "c:\wpwin60\macros\test.bif"

FORNEXT(x; 1; 5; 1)
 BIFWRITE(vStatus; "Group " + x; "Section " + x; "Item " + x; "Value " + x; WPWordString!; 0; vFile)
ENDFOR

ASSIGN(hBIF; APPLOCATE("BIF Edit*"))
IF(hBIF)
 APPACTIVATE(hBIF)
ELSE
 ASSIGN(vResult; APPEXECUTEEXT("c:\wpwin60\bifed20.exe"; 1))
ENDIF
IF(vResult > 31)
 SENDKEYS WAIT("{Alt + F}Oc:\wpwin60\macros\test.bif {Enter}{Enter}{Down}{Enter}{Down}")
 MESSAGEBOX(x; "BIFWRITE COMMAND"; "After you close the message box, double-click another" + HdRtrn + "Group, Section, or Item." + HdRtrn + HdRtrn + "Click Help for more information about BIF Edit."; IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Cannot execute BIFED20.EXE. Check the path and modify if necessary."; IconExclamation!)
 QUIT
ENDIF
```

---

## See Also

• BIFFILEPATH • BIFINFO • BIFREAD

---

# BREAK

---

## Purpose

Ends a loop, and directs macro execution to the first statement after the ending loop statement (see *Loop Statements* in *Macro Control Statements*). In a SWITCH statement, BREAK directs macro execution to the first statement after ENDSWITCH.

---

## Syntax

BREAK

---

## Description

BREAK bypasses the normal test expression of a loop or conditional statement. Statements after BREAK are ignored.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Declare and test variable types
// Demonstrate BREAK, EXIST, LOCAL, GLOBAL, and PERSIST commands

DISCARD var5; var1; var7
LOCAL var2; var9; var4
GLOBAL var3; var10; var6
PERSIST var5; var1; var7
ASSIGN(var8; "")

FOR(x; 1; x < 11; x + 1)
 ASSIGN(INDIRECT("var" + x); x)
 MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion! | YesNo!)
 IF(vStatus = 7)
 BREAK
 ENDIF
ENDFOR

FOR(z; 1; z < (x + 1); z + 1)
 SWITCH(EXISTS(INDIRECT("var" + z)))
 CASEOF 1: Msg("Local"; z)
 CASEOF 2: Msg("Global"; z)
 CASEOF 3: Msg("Persist"; z)
 ENDSWITCH
ENDFOR

FUNCTION Msg(w; y)
 MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w)
ENDFUNC
```

---

## See Also

♦ FOR ♦ FOREACH ♦ FORNEXT ♦ NEXT ♦ REPEAT ♦ WHILE ♦ SWITCH

---

# CALL

---

|                    |                                                                                       |
|--------------------|---------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Calls a subroutine that ends with RETURN (see Subroutines).                           |
| <b>Syntax</b>      | CALL ( <i>Label</i> )                                                                 |
| <b>Description</b> | RETURN directs macro execution to the statement that follows the subroutine's caller. |
| <b>Parameters</b>  |                                                                                       |

---

*Label* Label: A LABEL, PROCEDURE, or FUNCTION name that begins with a letter and consists of one or more letters or numbers. The format for calling LABEL is,

```
CALL(LabelSub)

...other statements...
```

```
LABEL(LabelSub)
...statement block...
RETURN
```

The formats for calling PROCEDURE and FUNCTION are,

```
CALL ProcSub(Parameter; Parameter)

...other statements...
```

```
PROCEDURE ProcSub(Parameter; Parameter)
...statementblock...
ENDPROC
```

```
CALL FuncSub(Parameter; Parameter)

...other statements...
```

```
FUNCTION FuncSub(Parameter; Parameter)
...statementblock...
ENDFUNC
```

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input for Y or N
// For CANCEL condition: chose Cancel, click Close, press Alt+F4, or double-click system menu box
// Demonstrate CALL command

CANCEL(On!)
ONCANCEL CALL(CancelMessage)
PROCEDURE Message(vPrompt; vIcon)
 IF(vIcon = "!")
 BEEP
 MESSAGEBOX(y; "Message Box"; vPrompt; IconExclamation!)
```

```

ELSE
 MESSAGEBOX(y; "Message Box"; vPrompt; IconInformation!)
ENDIF
ENDPROC

vStr := ""
REPEAT
 GETSTRING(vStr; "Press one character: ""Y"" for Yes or ""N"" for no"; "GETSTRING EXAMPLE";
 50)
 IF(vStr = "")
 vStr := "<Enter>"
 ENDIF

 vTest := TOUPPER(vStr)
 IF(CHARLEN(vStr) > 1)
 CALL Message("You typed or pressed "" + vStr + """"; "!")
 ELSE
 IF((vTest = "Y") OR (vTest = "N"))
 CALL Message("Well done - You pressed " + vTest; "i")
 ELSE
 CALL Message("You pressed "" + vStr + """"; "!")
 ENDIF
 ENDIF
UNTIL(((vTest = "Y") OR (vTest = "N")) AND NOT(CHARLEN(vStr) > 1))
QUIT

LABEL(CancelMessage)
vStr := "a Cancel control"
RETURN

```

---

## See Also

♦ CASE ♦ CASE CALL ♦ GO ♦ LABEL ♦ RETURN ♦ ♦

---

# CANCEL

|                    |                                                                                                                                                                                                                                                                                             |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Determines how a macro responds to a Cancel condition.                                                                                                                                                                                                                                      |
| <b>Syntax</b>      | CANCEL ( <i>State</i> )                                                                                                                                                                                                                                                                     |
| <b>Description</b> | Create a Cancel condition by pressing Esc, or with ASSERT(CancelCondition!).                                                                                                                                                                                                                |
| <b>Parameters</b>  |                                                                                                                                                                                                                                                                                             |
| <i>State</i>       | Enumeration: Specifies the Cancel state. The default is CANCEL(On!). The enumerations are, <ul style="list-style-type: none"> <li>Off! Overrides a Cancel condition.</li> <li>On! Stops a macro unless preceded by ONCANCEL, which directs macro execution to a specified LABEL.</li> </ul> |



## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input for Y or N
// For Cancel condition: chose Cancel, click Close, press Alt+F4, or double-click system menu box
// Demonstrate CANCEL command

CANCEL(On!)
ONCANCEL CALL(CancelMessage)
PROCEDURE Message(vPrompt; vlcon)
 IF(vlcon = "!")
 BEEP
 MESSAGEBOX(y; "Message Box"; vPrompt; IconExclamation!)
 ELSE
 MESSAGEBOX(y; "Message Box"; vPrompt; IconInformation!)
 ENDIF
ENDPROC

vStr := ""
REPEAT
 GETSTRING(vStr; "Press one character: ""Y"" for Yes or ""N"" for no"; "GETSTRING EXAMPLE";
 50)
 IF(vStr = "")
 vStr := "<Enter>"
 ENDIF

 vTest := TOUPPER(vStr)
 IF(CHARLEN(vStr) > 1)
 CALL Message("You typed or pressed "" + vStr + """"; "!")
 ELSE
 IF((vTest = "Y") OR (vTest = "N"))
 CALL Message("Well done - You pressed " + vTest; "i")
 ELSE
 CALL Message("You pressed "" + vStr + """"; "!")
 ENDIF
 ENDIF
UNTIL(((vTest = "Y") OR (vTest = "N")) AND NOT(CHARLEN(vStr) > 1))
QUIT

LABEL(CancelMessage)
vStr := "a Cancel control"
RETURN
```

---

## See Also

♦ ASSERT ♦ ERROR ♦ LABEL ♦ NOTFOUND ♦ ONCANCEL

---

# CASE

---

**Purpose**

A conditional statement that tests for matching expressions. If a match is found, a LABEL is executed (see *Conditional Statements* in *Macro Control Statements*).

**Syntax**

The general form of a CASE statement is,  
CASE (*Test*; {*Case*; *Label*; *Case*; *Label*...}; *DefaultLabel*)

**Description**

CASE compares *Test* to a set of *Cases* (values). If the first comparison is true (if *Test* and *Case* match), the *Label* following *Case* is executed. If the comparison is false, the next *Case* is evaluated and so forth. If no comparison is true, *DefaultLabel* is executed.

**Parameters**

*Test*

Any: The control expression. Variables are assigned values by commands such as GETSTRING, GETNUMBER, or MENU.

*Case*

Any: An expression (variable, constant, character) with a value that is usually assigned before the macro is compiled. It is possible to assign the value at run-time.

*Label*

Label: The LABEL to execute if *Case* matches *Test*.

*DefaultLabel*

Label: The LABEL to execute if no *Case* matches *Test*.

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Create a loop with GO command
// Demonstrate CASE, MESSAGEBOX, and system variables

LABEL(Start)
MENU(vChoice; Digit;;; {"IconExclamation!"; "IconInformation!"; "IconQuestion"; "IconStop!";
"Quit"})
CASE(vChoice; {1; Exclaim; 2; Info; 3; Question; 4; Stop; 5; QuitMacro}; QuitMacro)

LABEL(Exclaim)
MESSAGEBOX(x; "EXCLAMATION!"; "Example of an exclamation icon"; IconExclamation!)
GO(LoopToStart)

LABEL(Info)
ASSIGN(vDay; ?DateWeekday)
MESSAGEBOX(x; "INFORMATION!"; "Example of an information icon - The current day of the
week is " + vDay + ". "; IconInformation!)
GO(LoopToStart)

LABEL(Question)
Yes := 6
MESSAGEBOX(vAns; "QUESTION!"; "Example of a question icon. Would you like to know the
current document font?"; IconQuestion! | YesNo!)
IF(vAns = Yes)
vFont := ?DocumentFont
vPrompt := "The current document font is " + vFont
MESSAGEBOX(x; "FONT!"; vPrompt; IconInformation!)
```

```

 GO(LoopToStart)
ELSE
 GO(LoopToStart)
ENDIF

LABEL(Stop)
MESSAGEBOX(z; "STOP"; "Example of a stop icon"; IconStop!)
GO(LoopToStart)

LABEL(LoopToStart)
MESSAGEBOX(vAns; "Message"; "Would you like to continue?"; IconQuestion! | YesNo!)
IF(vAns = 7)
 GO(QuitMacro)
ELSE
 GO(Start)
ENDIF

LABEL(QuitMacro)
BEEP
QUIT

```

---

**See Also**

♦ CASE CALL ♦ IF ♦ LABEL ♦ SWITCH

---

## CASE CALL

---

**Purpose** A conditional statement that tests for matching expressions. If a match is found, a LABEL is called (see *Conditional Statements in Macro Control Statements*).

---

**Syntax** The general form of a CASE CALL statement is,  
CASE CALL (*Test*; {*Case*; *Label*; *Case*; *Label*...}; *DefaultLabel*)

---

**Description** CASE CALL compares *Test* to a set of *Cases* (values). If the first comparison is true (if *Test* and *Case* match), the *Label* following *Case* is executed. If the comparison is false, the next *Case* is evaluated and so forth. If no comparison is true, *DefaultLabel* is executed.

CASE is different from CASE CALL, in that a RETURN statement after LABEL directs macro execution to the statement that follows CASE CALL.

---

**Parameters**

*Test* Any: The control expression. Variables are assigned values by commands such as GETSTRING, GETNUMBER, or MENU.

|                     |                                                                                                                                                                        |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Case</i>         | Any: An expression (variable, constant, character) with a value that is usually assigned before the macro is compiled. It is possible to assign the value at run-time. |
| <i>Label</i>        | Label: The LABEL to execute if <i>Case</i> matches <i>Test</i> .                                                                                                       |
| <i>DefaultLabel</i> | Label: The LABEL to execute if no <i>Case</i> matches <i>Test</i> .                                                                                                    |

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Execute Windows Calculator, Calendar, and Cardfile
// Demonstrate CASE CALL inside a REPEAT-UNTIL loop

ASSIGN(hWP; APPLOCATE(""))
REPEAT
 MENU(vChoice; Digit;;; {"Calculator"; "Calendar"; "Card File"; "Quit"})
 CASE CALL(vChoice; {1; Program; 2; Program; 3; Program; 4; QuitMacro}; QuitMacro)
UNTIL(vChoice = 4)

LABEL(Program)
SWITCH(vChoice)
 CASEOF 1: vProgram := "c:\windows\calc.exe"
 CASEOF 2: vProgram := "c:\windows\calendar.exe"
 CASEOF 3: vProgram := "c:\windows\cardfile.exe"
ENDSWITCH
APPEXECUTE(vProgram)
RETURN

LABEL(QuitMacro)
MESSAGEBOX(vStatus; "EXIT"; "Are you finished?"; IconQuestion! ! YesNo!)
IF(vStatus = 6)
 APPACTIVATE(hWP)
 QUIT
ELSE
 vChoice := ""
 RETURN
ENDIF
```

### See Also

♦ CASE ♦ IF ♦ LABEL ♦ RETURN ♦ SWITCH

## CHAIN

### Purpose

Calls (starts) another macro when the parent macro ends.

### Syntax

CHAIN (*MacroFileName*)

### Description

A CHAIN macro does not return to its caller (see RUN). A macro must be compiled before it is called by CHAIN.

Although CHAIN can occur anywhere in a macro, it is the last statement to execute. You can cancel CHAIN (macro is not executed) with a QUIT statement. For example,

```
CHAIN(?Pathmacros + "Test4.wcm")

...other statements...

SWITCH(Test)
 CASEOF 1: CHAIN(?Pathmacros + "Test1.wcm")
 CASEOF 2: CHAIN(?Pathmacros + "Test2.wcm")
 CASEOF 3: CHAIN(?Pathmacros + "Test3.wcm")
 CASEOF 4: CALL(DoSomethingElse)
 DEFAULT: QUIT
ENDSWITCH
```

Explanation: If *Test* equals 1, 2, or 3, the corresponding macro plays when the parent macro ends. If *Test* equals 4, the DoSomethingElse subroutine is called and Test4.wcm plays when the parent macro ends. If *Test* does not equal 1, 2, 3, or 4, the macro quits and a CHAIN statement does not execute.

---

## Parameters

*MacroFileName*

Character Expression: The path and name of a compiled macro.

---

## Example

```
// There are four macros in this example
// Compile separately

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro1.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro2.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro3.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)

APPLICATION(A1; "WordPerfect"; Default; "US")
// Calling Macro
// Demonstrate CHAIN command and GLOBAL variables

GLOBAL vTitle; vPrompt
MENU(vChoice; Letter;; {"Macro 1"; "Macro 2"; "Macro 3"})
ASSIGN(vPrompt; "Computer beeps before chain macro called")
ASSIGN(vMacrosPath; ?PathMacros)
SWITCH(vChoice)
 CASEOF 1:
 ASSIGN(vTitle; "MACRO ONE")
 CHAIN(vMacrosPath + "macro1.wcm")
 CASEOF 2:
```

```

 ASSIGN(vTitle; "MACRO TWO")
 CHAIN(vMacrosPath + "macro2.wcm")
 CASEOF 3:
 ASSIGN(vTitle; "MACRO THREE")
 CHAIN(vMacrosPath + "macro3.wcm")
 ENDSWITCH
 // beep sounds before chain macro called
 BEEP WAIT(5)

```

## See Also

- ♦ RUN

# CHARLEN

|                    |                                                                                                                |
|--------------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Returns the number of characters in a string, including codes.                                                 |
| <b>Syntax</b>      | ReturnValue := CHARLEN ( <i>String</i> )                                                                       |
| <b>Returns</b>     | The number of characters, including codes, in a string.                                                        |
| <b>Description</b> | The string can be a variable, constant, character string, or result of an expression. See STRLEN for examples. |

## Parameters

*String* Character Expression: A variable, constant, character string, or result of an expression.

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract numbers from a character string
// Demonstrate CHARLEN command

vNumbers := ""
vPrompt := "Enter text that includes both letters and numbers:"

LABEL(Start)
GETSTRING(vStr; vPrompt; "EXTRACT NUMBERS")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vLen := CHARLEN(vStr)
vPos := 1
WHILE(NOT(vPos = vLen + 1))
 vTest := SUBCHAR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest
 ENDIF
 vPos := vPos + 1

```

```

ENDWHILE
IF(CHARLEN(vNumbers) > 0)
 MESSAGEBOX(x: "NUMBER STRING"; "You entered the following numbers: " + vNumbers;
 IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x: "ERROR"; "You didn't enter any numbers"; IconExclamation!)
ENDIF

```

---

**See Also**

♦ CHARPOS ♦ STRLEN ♦ SUBCHAR

---

## CHARPOS

**Purpose** Determines whether a character string is also a substring.

**Syntax** Return Value := CHARPOS (*String*; *SubString*)

**Returns** The beginning position of a substring, or zero if a substring is not found.

```
vPos := CHARPOS("WordPerfect"; "Perfect")
```

Result: vPos equals 5

```
vPos := CHARPOS("WordPerfect"; "Scott")
```

Result: vPos equals 0

---

**Parameters**

*String* Character Expression: A character string to evaluate.

*Substring* Character Expression: A substring to locate in *String*.

---

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract "Perfect" from WordPerfect
// Test if "Corporation" was entered
// Demonstrate CHARPOS command

vNumbers := ""
vPrompt := "Enter the following text: ""WordPerfect Corporation""

LABEL(Start)
GETSTRING(vStr; vPrompt; "USING SUBCHAR")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x: "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vSub := SUBCHAR(vStr; 5; 7)

```

```

IF(vSub = "Perfect")
 MESSAGEBOX(x; "SUBCHAR EXAMPLE"; "WordPerfect is simply "" + vSub + """;
 IconInformation!)
IF(CHARPOS(vStr; "Corporation") = 0)
 BEEP
 MESSAGEBOX(x; "CHARPOS EXAMPLE"; "You entered ""WordPerfect"" but not
 ""Corporation""; IconExclamation!)
ENDIF
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; ""WordPerfect"" wasn't the first word"; IconExclamation!)
 GO(Start)
ENDIF

```

## See Also

♦ CHARLEN ♦ STRPOS ♦ SUBCHAR

# CTON

## Purpose

Converts a keyboard character to its decimal equivalent.

## Syntax

ReturnValue := CTON (*Character*)

## Returns

The decimal equivalent of a keyboard character.

## Parameters

*Character*

Character Expression: A keyboard character to convert. If a string is used, the decimal equivalent of the first character is returned.

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract string of numbers and convert to numeric equivalent
// Demonstrate CTON command

vNumbers := ""
vPrompt := "Enter text that includes both letters and numbers:"
LABEL(Start)
GETSTRING(vStr; vPrompt; "EXTRACT NUMBERS")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vLen := STRLEN(vStr)
vPos := 1
REPEAT
 vTest := SUBSTR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest

```



```

ENDIF
vPos := vPos + 1
UNTIL(vPos = vLen + 1)
IF(STRLEN(vNumbers) > 0)
 vConvert := STRNUM(vNumbers)
 vConvert := vConvert * 5
 MESSAGEBOX(x, "STRNUM COMMAND"; vNumbers + " * 5 equals " + vConvert;
 IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x, "ERROR"; "You didn't enter any numbers"; IconExclamation!)
ENDIF

```

---

## See Also

♦ NTOC

---

# DDEEXECUTE

**Purpose** Sends a command string to a client application.

**Syntax** DDEEXECUTE (*ConversationID*; *Command*)

**Description** See the client application's documentation for command names and syntax.

## Parameters

*ConversationID* Numeric Expression: A number that links the server and client applications. DDEINITIATE returns the number in a variable.

*Command* Character Expression: A command to execute.

---

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Start conversation with WP Office 4.0a
// Execute Office 4.0a command
// Demonstrate DDEEXECUTE and DDEINITIATE commands

ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "COMMAND"))
IF(hConv) // IF hConv not equal to 0
 DDEEXECUTE(hConv; "ViewOpenNamed(ViewName:""Mail""; ViewType:Mail)")
 DDETERMINATEALL
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is WP Office running? Conversation not started."; IconQuestion!)
ENDIF

```

---

**See Also** ♦ DDEINITIATE ♦ DDEEXECUTEEXT

---

# DDEEXECUTEEXT

---

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>        | Sends a command to a client application, and returns a value that indicates if the command was successfully processed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Syntax</b>         | ReturnValue := DDEEXECUTEEXT ( <i>ConversationID</i> ; <i>Command</i> ; <i>TimeOut</i> ; <i>Label</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Returns</b>        | A number greater than 0 if successful, or 0 if not.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Description</b>    | See the client application's documentation for command names and syntax.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Parameters</b>     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>ConversationID</i> | Numeric Expression: A number that links the server and client applications. DDEINITIATE returns the number in a variable.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>Command</i>        | Character Expression: A command to execute.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <i>TimeOut</i>        | Numeric Expression: Asynchronous DDE: Set <i>TimeOut</i> to 0 or a negative value. DDE returns immediately, and notifies the macro that the command was delivered. When the client application acknowledges receiving the command, DDE notifies the macro and the macro calls <i>Label</i> . The return value is invalid for asynchronous DDE, because the client application has not yet processed the command.<br><br>Synchronous DDE: Set <i>TimeOut</i> to a positive integer, which specifies the maximum number of milliseconds to wait while the client application processes the command. The result is returned when the command is processed or DDE times out (command is not processed within the specified time). See <i>Return</i> above. |
| <i>Label</i>          | Label: The LABEL to call when the client application acknowledges receiving the command (see <i>TimeOut</i> parameter). If the LABEL statement does not include RETURN, macro behavior is unpredictable.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Start conversation with WP Office 4.0a
// Execute Office 4.0a command
// Demonstrate DDEEXECUTEEXT command

ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "COMMAND"))
IF(hConv) // IF hConv not equal to 0
 vReturn := DDEEXECUTEEXT(hConv; "ViewOpenNamed(""Mail""; Mail!); 0; Message)
 IF(vReturn = 0)
 MESSAGEBOX(x; "ERROR"; "Command not received by WP Office"; IconExclamation!)
 ENDIF
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is WP Office running? Conversation not started."; IconQuestion!)
ENDIF

DDETERMINATE(hConv)
QUIT
```

```

LABEL(Message)
 ASSIGN(vMsg; "WP Office acknowledges receipt of ViewOpenNamed command")
 MESSAGEBOX(z; "DDEEXECUTEEXT EXAMPLE"; vMsg; IconInformation!)
 RETURN

```

---

**See Also** ♦ DDEEXECUTE ♦ DDEINITIATE ♦ LABEL

---

## DDEINITIATE

---

**Purpose** Starts a DDE conversation with a client application.

---

**Syntax** DDEINITIATE (*ConversationID*; *ServiceName*, *TopicName*)

---

**Description** A conversation must be started between server and client applications before you can use commands such as DDEPOKE, DDEREQUEST, and DDEEXECUTE.

---

**Parameters**

*ConversationID* Variable: A number that links the server and client applications is returned in this variable, or zero if the conversation is not started.

*ServiceName* Character Expression: A DDE service name. See a client application's documentation.

*TopicName* Character Expression: A Topic name. For example, WordPerfect Office 4.0a for Windows has three topics names, depending on the type of action to initiate: COMMAND, GETOFFICEDATA and GETADDRESSBOOKDATA. See the client application's documentation.

---

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Start conversation with WP Office 4.0a
// Execute Office 4.0a command
// Demonstrate DDEEXECUTE and DDEINITIATE commands

ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "COMMAND"))
IF(hConv) // IF hConv not equal to 0
 DDEEXECUTE(hConv; "ViewOpenNamed(ViewName:""Mail""; ViewType:Mail!)")
 DDETERMINATEALL
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is WP Office running? Conversation not started."; IconQuestion!)
ENDIF

```

---

**See Also** ♦ DDEEXECUTE ♦ DDEPOKE ♦ DDEREQUEST ♦ DDETERMINATE ♦ DDETERMINATEALL

---

# DDEPOKE

---

**Purpose** Assigns data to an item in a client application.

---

**Syntax** DDEPOKE (*ConversationID*; *ItemName*; *ItemData*)

---

## Parameters

*ConversationID* Numeric Expression: A number that links the server and client applications. DDEINITIATE returns the number in a variable.

*ItemName* Character expression: A client application item. See the client application's documentation for item names.

*ItemData* Character expression: The data to assign to a client application item.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Assign fax number to ExpressFax+
// Demonstrate DDEPOKE command

ASSIGN(hConv; DDEINITIATE("FAXMNG"; "Transmit"))
IF(hConv)
 GETSTRING(vFax; "Enter FAX Number"; "ExpressFax+")
 DDEPOKE(hConf; "FAX Number"; vFax)
 DDETERMINATE(hConv)
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is ExpressFax+ running? Conversation not started."; IconQuestion!)
ENDIF
```

---

**See Also** ♦ DDEINITIATE ♦ DDEREQUEST

---

---

# DDEREQUEST

---

**Purpose** Assigns the contents of a client application item to a macro variable.

---

**Syntax** ReturnValue := DDEREQUEST (*ConversationID*; *ItemName*)

---

**Returns** The contents of the client application item specified by *ItemName*.

---

## Parameters

*ConversationID* Numeric Expression: A number that links the server and client applications. DDEINITIATE returns the number in a variable.

*ItemName*

Character expression: The name of a client application item.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Request WP Office version number
// Demonstrate DDEREQUEST command

ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "GETOFFICEDATA"))
IF(hConv)
 ASSIGN(vMajor; DDEREQUEST(hConv; "GetOfficeData(ID; MajorVersion!)))
 ASSIGN(vMinor; DDEREQUEST(hConv; "GetOfficeData(ID; MinorVersion!)))
 MESSAGEBOX(x; "DDEREQUEST EXAMPLE"; "WP Office version is " + vMajor + "." + vMinor;
 IconInformation!)
 DDETERMINATE(hConv)
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is WP Office running? Conversation not started."; IconQuestion!)
ENDIF
```

---

**See Also**

♦ DDEINITIATE ♦ DDEPOKE ♦ ONDDEADVISE CALL

---

## DDETERMINATE

---

**Purpose**

Ends a DDE conversation with a client application.

---

**Syntax**

DDETERMINATE (*ConversationID*)

---

**Parameters**

*ConversationID*

Numeric Expression: A number that links the server and client applications. DDEINITIATE returns the number in a variable.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Request WP Office version number
// Demonstrate DDETERMINATE command

ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "GETOFFICEDATA"))
IF(hConv)
 ASSIGN(vMajor; DDEREQUEST(hConv; "GetOfficeData(ID; MajorVersion!)))
 ASSIGN(vMinor; DDEREQUEST(hConv; "GetOfficeData(ID; MinorVersion!)))
 MESSAGEBOX(x; "DDEREQUEST EXAMPLE"; "WP Office version is " + vMajor + "." + vMinor;
 IconInformation!)
 DDETERMINATE(hConv)
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is WP Office running? Conversation not started."; IconQuestion!)
ENDIF
```

---

**See Also**♦ DDEINITIATE ♦ DDETERMINATEALL

---

---

## DDETERMINATEALL

---

**Purpose** Ends all DDE conversations.**Syntax** DDETERMINATEALL**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Start conversation with WP Office 4.0a
// Execute Office 4.0a command
// Demonstrate DDETERMINATE commands

ASSIGN(hConv; DDEINITIATE("WPOFFICE40"; "COMMAND"))
IF(hConv // IF hConv not equal to 0
 DDEEXECUTE(hConv; "ViewOpenNamed(ViewName:""Mail""; ViewType:Mail!)")
 DDETERMINATEALL
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Is WP Office running? Conversation not started."; IconQuestion!)
ENDIF
```

---

**See Also**♦ DDEINITIATE ♦ DDETERMINATE

---

---

## DECLARE

---

**Purpose** Creates a local variable or array, which stores as many variables, constants, or expressions as memory allows.**Syntax** DECLARE (*VariableName*[*Elements*]; ...)**Description** An array can have up to 10 dimensions, and up to 32,767 elements per dimension depending on the amount of available memory. In the following example, DECLARE creates ArrayA with ten elements and ArrayB with five elements. An element in each array is assigned a value. Parentheses are optional.

```
DECLARE(ArrayA[10]; ArrayB[5])
ArrayA[3] := "John"
```

Explanation: Assigns "John" to the third element in ArrayA

```
ArrayB[5] := ArrayA[3]
```

Explanation: Assigns "John" to the fifth element in ArrayB.

In the next example, DECLARE creates a two-dimensional array (ArrayC) with 100 elements (ten rows and ten columns). One element is assigned a value.

```
DECLARE ArrayC[10;10]
ArrayC[2;3] := 3+6
```

Explanation: Assigns 9 to the element at row 2, column 3.

---

## Parameters

### *VariableName*

Variable: The name of a local variable or array. Use a semicolon to separate names when declaring more than one variable or array.

### *Elements*

Numeric Expression: The number of elements in an array, enclosed in brackets. Use a semicolon to separate elements in a multi-dimensional array.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Declare an array of months
// Demonstrate DECLARE command, recursive LABEL statement

DECLARE Months[12]
x := 1

FOREACH(vMonth; {"January"; "February"; "March"; "April"; "May"; "June"; "July"; "August";
"September"; "October"; "November"; "December"})
 Months[x] := vMonth
 x := x + 1
ENDFOR

LABEL(Start)
MENU(vNbr; Letter;;; {"January"; "February"; "March"; "April"; "May"; "June"; "July"; "August";
"September"; "October"; "November"; "December"; "QUIT"})
IF(vNbr = 13)
 BEEP
 QUIT
ELSE
 MESSAGEBOX(vStatus; "MONTHS OF THE YEAR"; "You selected " + Months[vNbr];
IconInformation!)
 GO(Start)
ENDIF
```

---

## See Also

♦ GLOBAL ♦ LOCAL ♦ PERSIST ♦ PERSISTALL

---

# DEFAULTUNITS

---

**Purpose** DEFAULTUNITS determines the unit of measure for commands that do not specify a unit of measure.

---

**Syntax** DEFAULTUNITS (*Units*)

---

**Description** In the first example, Advance uses the default unit of measure (WP Units). In the second, Advance overrides the default unit of measure and uses inches. In the third, DEFAULTUNITS specifies centimeters as the default unit of measure.

Advance(AdvanceFromTop!; 4)

Explanation: Advance insertion point 4 WP Units.

Advance (AdvanceFromTop!; 4")

Explanation: Advance insertion point 4 inches.

DEFAULTUNITS(Inches!)

Advance(AdvanceFromTop!; 4")

Explanation: Advance insertion point 4 inches.

The default unit of measure is effective until changed by another DEFAULTUNITS command. If not specified by DEFAULTUNITS, the default unit of measure is WP Units.

---

## Parameters

*Units*

Enumeration: A unit of measure. The enumerations are,

Inches!

Centimeters!

Millimeters!

Points!

One Point equals 1/72 of an inch.

WPUnits!

One WordPerfect unit equals 1/1200 of an inch.

WP1200ths!

Synonym for WPUnits!.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
```

```
// Specify a unit of measure or accept WP units default
```

```
// Demonstrate DEFAULTUNITS and GETUNITS commands
```

```
MENU(vChoice; Digit;;; {"Advance down"; "Advance up"; "Advance left"; "Advance right"})
```

```
SWITCH(vChoice)
```

```
 CASEOF 1: vPrompt := "Advance down:"
```

```
 CASEOF 2: vPrompt := "Advance up:"
```

```
 CASEOF 3: vPrompt := "Advance left:"
```

```
 CASEOF 4: vPrompt := "Advance right:"
```

```
ENDSWITCH
```

```
DEFAULTUNITS(Inches!)
```

```
GETUNITS(vUnit; vPrompt; "GETUNITS EXAMPLE")
```

```
SWITCH(vChoice)
```



```

CASEOF 1: ADVANCE(AdvanceDown!; vUnit)
CASEOF 2: ADVANCE(AdvanceUp!; vUnit)
CASEOF 3: ADVANCE(AdvanceLeft!; vUnit)
CASEOF 4: ADVANCE(AdvanceRight!; vUnit)
ENDSWITCH

```

---

## DIALOGADDCHECKBOX

|                   |                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Adds a check box control to a dialog box.                                                                                                                                                                                                                       |
| <b>Syntax</b>     | DIALOGADDCHECKBOX ( <i>Dialog; Control; Left; Top; Width; Height; Text; MacroVar</i> )                                                                                                                                                                          |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                 |
| <i>Dialog</i>     | Character Expression: The name or number of the dialog box to contain the check box control (see DIALOGDEFINE).                                                                                                                                                 |
| <i>Control</i>    | Character Expression: A name or number that identifies the check box control (see DIALOGDISPLAY).                                                                                                                                                               |
| <i>Left</i>       | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the check box.                                                                                                                                              |
| <i>Top</i>        | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the check box.                                                                                                                                                          |
| <i>Width</i>      | Numeric Expression: Width of the check box control in dialog units.                                                                                                                                                                                             |
| <i>Height</i>     | Numeric Expression: Height of the check box control in dialog units.                                                                                                                                                                                            |
| <i>Text</i>       | Character Expression: Text displayed to the right of the check box.                                                                                                                                                                                             |
| <i>MacroVar</i>   | Variable: 1 is returned in this variable when check box is checked, and 0 when not checked. The default value is 0. If <i>MacroVar</i> is assigned a value greater than 0 before the macro compiles, the check box is checked when the dialog box is displayed. |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> APPLICATION(A1; "WordPerfect"; Default; "US") // Demonstrate UNITSTR and DIALOGADDCHECKBOX commands  vNbr := 1 vStatus := 4 DIALOGDEFINE(1000; 50; 50; 100; 120; Percent! !NoTitle! !NoFrame!; "")   DIALOGADDTTEXT(1000; -1; 10; 15; 75; 15; Left!; "Convert " + vNbr + " inch to:")   DIALOGADDCHECKBOX(1000; 101; 15; 30; 100; 13; "Centimeters"; vCB1)   DIALOGADDCHECKBOX(1000; 102; 15; 45; 100; 13; "Millimeters"; vCB2)   DIALOGADDCHECKBOX(1000; 103; 15; 60; 100; 13; "Points"; vCB3)   DIALOGADDCHECKBOX(1000; 104; 15; 75; 100; 13; "WP Units"; vCB4)   DIALOGADDTTEXT(1000; -2; 10; 100; 100; 15; Left!; "(Press Alt+F4 to cancel)") DIALOGDISPLAY(1000; -1; Msg) </pre> |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

WHILE(vStatus = 4)
ENDWHILE
DIALOGDESTROY(1000)

LABEL(Msg)
 IF(Msg[5] = 274)
 BEEP
 vStatus := 2
 ENDIF
 DEFAULTUNITS(Inches!)
 SWITCH(Msg[3])
 CASEOF 101: vResult := UNITSTR(vNnbr; Centimeters!)
 Message
 CASEOF 102: vResult := UNITSTR(vNnbr; Millimeters!)
 Message
 CASEOF 103: vResult := UNITSTR(vNnbr; Points!)
 Message
 CASEOF 104: vResult := UNITSTR(vNnbr; WPUnits!)
 Message
 ENDSWITCH
 RETURN

LABEL(Message)
 DIALOGUNDISPLAY(1000; "CancelBbtn")
 MESSAGEBOX(vStatus; "UNITSTR COMMAND"; vNnbr + " inch is converted to " + vResult;
 IconInformation! | RetryCancel!)
 IF(vStatus = 4)
 DIALOGDISPLAY(1000; -1; Msg)
 ELSE
 BEEP
 ENDIF
 RETURN

```

---

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

## DIALOGADDCOLORWHEEL

---

**Purpose**

Adds a color wheel control to a dialog box, and returns a number that represents the selected color.

---

**Syntax**

DIALOGADDCOLORWHEEL (*Dialog; Control; Left; Top; Width; Height; MacroVar*)

---

**Description**

The color selection consists of three color elements (red, green, and blue), separated into color values by the formula given below (see *MacroVar* parameter). These values are used in Windows programming to specify the color of a pen or brush.

---

## Parameters

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>   | Character Expression: The name or number of the dialog box to contain the color wheel control (see DIALOGDEFINE).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>Control</i>  | Character Expression: A name or number that identifies the color wheel control (see DIALOGDISPLAY).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>Left</i>     | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the color wheel control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>Top</i>      | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the color wheel control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>Width</i>    | Numeric Expression: Width of the color wheel control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>Height</i>   | Numeric Expression: Height of the color wheel control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>MacroVar</i> | Variable: A number corresponding to a color selection is returned in this variable. This number can be divided into three other numbers which correspond to red, green, and blue. The formulas are,<br>$vRed := MacroVar \% 256$ (vRed contains the red component, which is the remainder of <i>MacroVar</i> divided by 256)<br>$vGreen := MacroVar / 256 \% 256$ (vGreen contains the green component, which is the remainder of <i>MacroVar</i> divided by 256 divided by 256)<br>$vBlue := MacroVar / 256 / 256$ (vBlue contains the blue component, which is the result of <i>MacroVar</i> divided by 256 divided by 256) |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDCHECKBOX, DIALOGADDCOLORWHEEL,
DIALOGADDCOMBOBOX, and DIALOGADDCOUNTER

CALL(DlgCreate)
REPEAT
 MENU(vResult; Digit;;; {"Check box"; "Color wheel"; "Combination box"; "Counter"; "Quit"})
 DisplayControls
UNTIL(vResult = 5)
CALL(QuitMacro)

LABEL(DisplayControls)
x := 0
SWITCH(vResult)
 CASEOF 1: DIALOGDISPLAY(1000; 100; Msg)
 WHILE(x = 0)
 ENDWHILE
 CASEOF 2: DIALOGDISPLAY(1001; 1)
 vRed := vCW \% 256
 vGreen := vCW / 256 \% 256
 vBlue := vCW / 256 / 256
```

```

 DisplayColorValues(vRed; vGreen; vBlue)
 CASEOF 3: DIALOGDISPLAY(1002; 100)
 CASEOF 4: DIALOGDISPLAY(1003; 100)
 ENDSWITCH
 RETURN

LABEL(Msg)
 IF(Msg[5] = 274)
 DIALOGUNDISPLAY(1000; "CancelBtn")
 x := 1
 ENDIF
 SWITCH(Msg[3])
 CASEOF 100:
 DIALOGUNDISPLAY(1000; "CancelBtn")
 MESSAGEBOX(x; "SEE WHAT YOU CAN DO"; "This check box is used with a callback
 function. See DIALOGDISPLAY for details."; IconInformation!)
 ENDSWITCH
 RETURN

PROCEDURE DisplayColorValues(r; g; b)
 MESSAGEBOX(x; "COLOR VALUES"; "The red component equals " + r + ". The green component
 equals " + g + ". The blue component equals " + b + "."; IconInformation!)
ENDPROC

LABEL(DlgCreate)
 DIALOGDEFINE (1000; 50; 50; 120; 50; Percent! | NoFrame! "DIALOGADDCHECKBOX")
 DIALOGADDCHECKBOX(1000; 100; 10; 10; 100; 15; "Click check box"; vCB)
 DIALOGDEFINE (1001; 50; 50; 125; 130; Percent! | NoFrame! | OK!;
 "DIALOGADDCOLORWHEEL")
 DIALOGADDCOLORWHEEL(1001; 101; 10; 10; 105; 80; vCW)
 DIALOGDEFINE (1002; 50; 50; 125; 115; Percent! | NoFrame! "DIALOGADDCOMBOBOX")
 DIALOGADDCOMBOBOX(1002; 100; 10; 10; 100; 50; DropDown!; var)
 DIALOGADDLISTITEM(1002; 100; "DropDown! combo box")
 DIALOGADDCOMBOBOX(1002; 101; 10; 40; 100; 50; Simple!; var)
 DIALOGADDLISTITEM(1002; 101; "Simple! combo box")
 DIALOGDEFINE (1003; 50; 50; 120; 50; Percent! | NoFrame! "DIALOGADDCOUNTER")
 var := 5
 DIALOGADDCOUNTER(1003; 100; 10; 10; 10; 50; 15; 0; var; 1; 10; 1)
 RETURN

LABEL(QuitMacro)
 FORNEXT(x; 1000; 4; 1)
 DIALOGDESTROY(x)
 ENDFOR
 QUIT

```

---

## See Also

♦ DIALOGDEFINE ♦ DIALOGDISPLAY

# DIALOGADDCOMBOBOX

---

|                    |                                                                                                                                                             |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Adds a combo box control to a dialog box.                                                                                                                   |
| <b>Syntax</b>      | DIALOGADDCOMBOBOX ( <i>Dialog; Control; Left; Top; Width; Height; Style; MacroVar; LimitText</i> )                                                          |
| <b>Description</b> | Displays an edit box and a list box. You enter text in the edit box, or double-click a list item to insert. Use DIALOGADDLISTITEM to create list box items. |

---

## Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|---------|------------------------------------------------------------|---|-------|-----------------------------------------------------------------------------------------------------------------------------|---|-----------|--------------------------------------------------------------------------------------------------------------|----|-----------|-----------------------------------------------------------------------------------------------------------------------|----|----------|------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the combo box control (see DIALOGDEFINE).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| <i>Control</i> | Character Expression: A name or number that identifies the combo box control (see DIALOGDISPLAY).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the combo box.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the combo box.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| <i>Width</i>   | Numeric Expression: Width of the combo box control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| <i>Height</i>  | Numeric Expression: Height of the combo box control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                             |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| <i>Style</i>   | Enumeration: Combo box styles. Type   between enumerations to combine styles. The values, enumerations, and descriptions are,<br><table><tr><td>0</td><td>Simple!</td><td>Displays an edit control and an opened drop-down list box.</td></tr><tr><td>1</td><td>Sort!</td><td>Items are sorted alphabetically. If Sort! is used by itself, Simple! is the default combo box style. See DIALOGADDLISTITEM.</td></tr><tr><td>8</td><td>Dropdown!</td><td>Displays an edit control. Clicking the arrow to the right of the edit control displays a drop-down list box.</td></tr><tr><td>16</td><td>Droplist!</td><td>Displays a read-only edit control. Clicking the arrow to the right of the edit control displays a drop-down list box.</td></tr><tr><td>32</td><td>WPChars!</td><td>Non-keyboard characters allowed in a list item. See DIALOGADDLISTITEM.</td></tr></table> | 0                                                                                                                           | Simple! | Displays an edit control and an opened drop-down list box. | 1 | Sort! | Items are sorted alphabetically. If Sort! is used by itself, Simple! is the default combo box style. See DIALOGADDLISTITEM. | 8 | Dropdown! | Displays an edit control. Clicking the arrow to the right of the edit control displays a drop-down list box. | 16 | Droplist! | Displays a read-only edit control. Clicking the arrow to the right of the edit control displays a drop-down list box. | 32 | WPChars! | Non-keyboard characters allowed in a list item. See DIALOGADDLISTITEM. |
| 0              | Simple!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Displays an edit control and an opened drop-down list box.                                                                  |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| 1              | Sort!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Items are sorted alphabetically. If Sort! is used by itself, Simple! is the default combo box style. See DIALOGADDLISTITEM. |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| 8              | Dropdown!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Displays an edit control. Clicking the arrow to the right of the edit control displays a drop-down list box.                |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| 16             | Droplist!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Displays a read-only edit control. Clicking the arrow to the right of the edit control displays a drop-down list box.       |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |
| 32             | WPChars!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Non-keyboard characters allowed in a list item. See DIALOGADDLISTITEM.                                                      |         |                                                            |   |       |                                                                                                                             |   |           |                                                                                                              |    |           |                                                                                                                       |    |          |                                                                        |

### Style combination examples

Sort! | Simple!

Explanation: Name search allowed on a sorted list box. A vertical scroll bar automatically appears when needed.

*MacroVar*

Variable: The selected combo box item is returned in this variable. An initial item can be assigned to *MacroVar* before the macro compiles.

*LimitText*

Numeric Expression: Maximum number of characters the edit control accepts, up to the number of characters that fit in the edit box. This number has no effect if you choose style 16 (DropDown3D!).

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDCHECKBOX, DIALOGADDCOLORWHEEL,
DIALOGADDCOMBOBOX, and DIALOGADDCOUNTER

CALL(DlgCreate)
REPEAT
 MENU(vResult; Dlg; { "Check box"; "Color wheel"; "Combination box"; "Counter"; "Quit" })
 DisplayControls
UNTIL(vResult = 5)
CALL(QuitMacro)

LABEL(DisplayControls)
x := 0
SWITCH(vResult)
CASEOF 1: DIALOGDISPLAY(1000; 100; Msg)
 WHILE(x = 0)
 ENDWHILE
CASEOF 2: DIALOGDISPLAY(1001; 1)
 vRed := vCW % 256
 vGreen := vCW / 256 % 256
 vBlue := vCW / 256 / 256
 DisplayColorValues(vRed; vGreen; vBlue)
CASEOF 3: DIALOGDISPLAY(1002; 100)
CASEOF 4: DIALOGDISPLAY(1003; 100)
ENDSWITCH
RETURN

LABEL(Msg)
IF(Msg[5] = 274)
 DIALOGUNDISPLAY(1000; "CancelBtn")
 x := 1
ENDIF
SWITCH(Msg[3])
CASEOF 100:
 DIALOGUNDISPLAY(1000; "CancelBtn")
 MESSAGEBOX(x; "SEE WHAT YOU CAN DO"; "This check box is used with a callback
function. See DIALOGDISPLAY for details."; IconInformation!)
ENDSWITCH
RETURN

PROCEDURE DisplayColorValues(r; g; b)
MESSAGEBOX(x; "COLOR VALUES"; "The red component equals " + r + ". The green component
equals " + g + ". The blue component equals " + b + "."; IconInformation!)
ENDPROC
```

```

LABEL(DlgCreate)
 DIALOGDEFINE (1000; 50; 50; 120; 50; Percent! | NoFrame!; "DIALOGADDCHECKBOX")
 DIALOGADDCHECKBOX(1000; 100; 10; 10; 100; 15; "Click check box"; vCB)
 DIALOGDEFINE (1001; 50; 50; 125; 130; Percent! | NoFrame! | OK!;
 "DIALOGADDCOLORWHEEL")
 DIALOGADDCOLORWHEEL(1001; 101; 10; 10; 105; 80; vCW)
 DIALOGDEFINE (1002; 50; 50; 125; 115; Percent! | NoFrame!; "DIALOGADDCOMBOBOX")
 DIALOGADDCOMBOBOX(1002; 100; 10; 10; 100; 50; DropDown!; var)
 DIALOGADDLISTITEM(1002; 100; "DropDown! combo box")
 DIALOGADDCOMBOBOX(1002; 101; 10; 40; 100; 50; Simple!; var)
 DIALOGADDLISTITEM(1002; 101; "Simple! combo box")
 DIALOGDEFINE (1003; 50; 50; 120; 50; Percent! | NoFrame!; "DIALOGADDCOUNTER")
 var := 5
 DIALOGADDCOUNTER(1003; 100; 10; 10; 50; 15; 0; var; 1; 10; 1)
 RETURN

LABEL(QuitMacro)
 FORNEXT(x; 1000; 4; 1)
 DIALOGDESTROY(x)
 ENDFOR
 QUIT

```

---

## See Also

- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGADDLISTBOX
- ♦ DIALOGADDLISTITEM ♦ DIALOGADDDPOPUPBUTTON

---

# DIALOGADDCONTROL

|                   |                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Adds a custom control to a dialog box.                                                                                  |
| <b>Syntax</b>     | DIALOGADDCONTROL ( <i>Dialog; Control; Left; Top; Width; Height; Class; Style; WindowName; MacroVar; Instance</i> )     |
| <b>Parameters</b> |                                                                                                                         |
| <i>Dialog</i>     | Character Expression: The name or number of the dialog box to contain the custom control (see DIALOGDEFINE).            |
| <i>Control</i>    | Character Expression: A name or number that identifies the custom control (see DIALOGDISPLAY).                          |
| <i>Left</i>       | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the custom control. |
| <i>Top</i>        | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the custom control.             |

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Width</i>      | Numeric Expression: Width of the custom control in dialog units.                                                                                                                                                                                                                                                                                                                                                                  |
| <i>Height</i>     | Numeric Expression: Height of the custom control in dialog units.                                                                                                                                                                                                                                                                                                                                                                 |
| <i>Class</i>      | Character Expression: The name of the control class. Standard Windows classes include Button, ComboBox, Edit, ListBox, ScrollBar, and Static. Other controls such as Meters and Spin buttons must be defined in a DLL (see <i>Instance</i> parameter). <i>Class</i> names for WordPerfect custom controls are in the TKSHWIN.H file on the diskette that comes with <i>WordPerfect 6.0 for Windows Software Developer's Kit</i> . |
| <i>Style</i>      | Numeric Expression: A number that defines a class style. For example, an edit control includes styles such as left, multiline, and password. The values for standard Windows styles are in the WINDOWS.H file.                                                                                                                                                                                                                    |
| <i>WindowName</i> | Character Expression: Control text. For example, the text that appears on a button control. Set this value to "" if the control does not accept text.                                                                                                                                                                                                                                                                             |
| <i>MacroVar</i>   | Variable: The value returned by a control. For example, a check box returns 1 if checked, and 0 if not checked.                                                                                                                                                                                                                                                                                                                   |
| <i>Instance</i>   | Numeric Expression: The handle of the DLL that contains the custom control. DLLLOAD returns the handle of a custom DLL in a variable. The Shared Code default is 0 (see <i>WordPerfect 6.0 for Windows Software Developer's Kit</i> for information about custom controls).                                                                                                                                                       |

---

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDCONTROL and SENDKEYS commands

InhibitInput(Off!)
WM_SYSCOMMAND := 274
z := 0
DIALOGDEFINE (1000; 50; 50; 122; 150; Percent! | NoFrame! | OK!; "DIALOGADDCONTROL")
 DIALOGADDCHECKBOX(1000; 100; 10; 10; 75; 15; "Check for demo"; vCB)
 DIALOGADDCONTROL(1000; 101; 10; 40; 100; 75; "wpnslb20"; 2; ""; vListItem; 0)
 A := "Apple"
 B := "Apple pie"
 C := "Applesauce"
 D := "Banana"
 E := "Banana oil"
 F := "Banana seat"
 G := "Banana split"
 FORNEXT(x; 65; 71; 1)
 DIALOGADDLISTITEM(1000; 101; INDIRECT(NTOC(x)))
 ENDFOR
DIALOGDISPLAY(1000; 101; Msg)
WHILE(z = 0)
ENDWHILE
DIALOGDESTROY(1000)

LABEL(Msg)
 IF(Msg[5] = WM_SYSCOMMAND)
 z := 1
 ENDIF

```



```

SWITCH(Msg[3])
 CASEOF 100: Demo
 CASEOF "OKBtn": DIALOGUNDISPLAY(1000; 101)
 IF(STRLEN(vListItem) > 0)
 ASSIGN(vSelect; "You selected "" + vListItem + "" from the list box")
 ELSE
 ASSIGN(vSelect; "You didn't select an item from the list box")
 ENDIF
 MESSAGEBOX(x; "LIST ITEM"; vSelect; IconInformation!)
 z := 1
 ENDSWITCH
 RETURN

LABEL(Demo)
 SENDKEYS WAIT("{Tab} {Up}")
 FOREACH(vLetter; {"A"; "p"; "p"; "l"; "e"; "s"; "a"; "u"; "c"; "e"})
 SENDKEYS WAIT(vLetter)
 WAIT(3)
 ENDFOR
 WAIT(12)
 FORNEXT(w; 1; 10; 1)
 SENDKEYS WAIT("{BackSpace}")
 WAIT(1)
 ENDFOR
 WAIT(10)
 FOREACH(vLetter; {"B"; "a"; "n"; "a"; "n"; "a"; " "; "s"; "p"; "l"; "i"; "t"})
 SENDKEYS WAIT(vLetter)
 WAIT(3)
 ENDFOR
 WAIT(12)
 SENDKEYS WAIT("{Tab}")
 MESSAGEBOX(x; "NAME SEARCH LIST BOX"; "Enter a word from the list to experiment with
 WordPerfect's custom list box")
 SENDKEYS WAIT("{Up}")
 RETURN

```

---

## See Also

• DIALOGDEFINE • DIALOGDISPLAY

---

# DIALOGADDCOUNTER

---

## Purpose

Adds a counter control to a dialog box.

---

## Syntax

DIALOGADDCOUNTER (*Dialog; Control; Left; Top; Width; Height; Format; MacroVar; CountMin; CountMax; CountStep*)

|                    |                                                                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b> | Click the counter button to insert a number in the edit box that is within a specified range (see <i>CountMin</i> and <i>CountMax</i> parameters).                          |
| <b>Parameters</b>  |                                                                                                                                                                             |
| <i>Dialog</i>      | Character Expression: The name or number of the dialog box to contain the counter control (see DIALOGDEFINE).                                                               |
| <i>Control</i>     | Character Expression: A name or number that identifies the counter control (see DIALOGDISPLAY).                                                                             |
| <i>Left</i>        | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the counter control.                                                    |
| <i>Top</i>         | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the counter control.                                                                |
| <i>Width</i>       | Numeric Expression: Width of the counter control in dialog units.                                                                                                           |
| <i>Height</i>      | Numeric Expression: Height of the counter control in dialog units.                                                                                                          |
| <i>Format</i>      | Numeric Expression: Reserved for future use. Set parameter to 0.                                                                                                            |
| <i>MacroVar</i>    | Variable: The selected counter number is returned in this variable. An initial number can be assigned to <i>MacroVar</i> before the macro compiles.                         |
| <i>CountMin</i>    | Numeric Expression: The minimum number that you can insert by clicking the counter button. You can enter a number in the edit control beyond the minimum and maximum range. |
| <i>CountMax</i>    | Numeric Expression: The maximum number that you can insert by clicking the counter button. You can enter a number in the edit control beyond the minimum and maximum range. |
| <i>CountStep</i>   | Numeric Expression: The increment and decrement step value.                                                                                                                 |

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDCHECKBOX, DIALOGADDCOLORWHEEL,
DIALOGADDCOMBOBOX, and DIALOGADDCOUNTER

CALL(DlgCreate)
REPEAT
 MENU(vResult; Digit;; {"Check box"; "Color wheel"; "Combination box"; "Counter"; "Quit"})
 DisplayControls
UNTIL(vResult = 5)
CALL(QuitMacro)

LABEL(DisplayControls)
x := 0
SWITCH(vResult)
CASEOF 1: DIALOGDISPLAY(1000; 100; Msg)
WHILE(x = 0)
ENDWHILE
CASEOF 2: DIALOGDISPLAY(1001; 1)
vRed := vCW % 256
```

```

vGreen := vCW / 256 % 256
vBlue := vCW / 256 / 256
DisplayColorValues(vRed; vGreen; vBlue)
CASEOF 3: DIALOGDISPLAY(1002; 100)
CASEOF 4: DIALOGDISPLAY(1003; 100)
ENDSWITCH
RETURN

LABEL(Msg)
IF(Msg[5] = 274)
 DIALOGUNDISPLAY(1000; "CancelBtn")
 x := 1
ENDIF
SWITCH(Msg[3])
CASEOF 100:
 DIALOGUNDISPLAY(1000; "CancelBtn")
 MESSAGEBOX(x; "SEE WHAT YOU CAN DO"; "This check box is used with a callback
function. See DIALOGDISPLAY for details."; IconInformation!)
ENDSWITCH
RETURN

PROCEDURE DisplayColorValues(r; g; b)
MESSAGEBOX(x; "COLOR VALUES"; "The red component equals " + r + ". The green component
equals " + g + ". The blue component equals " + b + "."; IconInformation!)
ENDPROC

LABEL(DlgCreate)
DIALOGDEFINE (1000; 50; 50; 120; 50; Percent! | NoFrame! | "DIALOGADDCHECKBOX")
 DIALOGADDCHECKBOX(1000; 100; 10; 10; 100; 15; "Click check box"; vCB)
DIALOGDEFINE (1001; 50; 50; 125; 130; Percent! | NoFrame! | OK!;
"DIALOGADDCOLORWHEEL")
 DIALOGADDCOLORWHEEL(1001; 101; 10; 10; 105; 80; vCW)
DIALOGDEFINE (1002; 50; 50; 125; 115; Percent! | NoFrame!; "DIALOGADDCOMBOBOX")
 DIALOGADDCOMBOBOX(1002; 100; 10; 10; 100; 50; DropDown!; var)
 DIALOGADDLISTITEM(1002; 100; "DropDown! combo box")
 DIALOGADDCOMBOBOX(1002; 101; 10; 40; 100; 50; Simple!; var)
 DIALOGADDLISTITEM(1002; 101; "Simple! combo box")
DIALOGDEFINE (1003; 50; 50; 120; 50; Percent! | NoFrame!; "DIALOGADDCOUNTER")
 var := 5
 DIALOGADDCOUNTER(1003; 100; 10; 10; 50; 15; 0; var; 1; 10; 1)
RETURN

LABEL(QuitMacro)
FORNEXT(x; 1000; 4; 1)
 DIALOGDESTROY(x)
ENDFOR
QUIT

```

---

## See Also

♦ DEFAULTUNITS ♦ DIALOGDEFINE

---

# DIALOGADDEDITBOX

---

**Purpose** Adds an edit control to a dialog box.

---

**Syntax** DIALOGADDEDITBOX (*Dialog; Control; Left; Top; Width; Height; Style; MacroVar; LimitText*)

---

**Description** The text entered in the edit box is returned in *MacroVar* (see *MacroVar* parameter).

---

## Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------------------|---|--------|-----------------------------------------------------------------------------------|---|---------|----------------------------------------------------------------------------|---|----------|------------------------------------------|----|----------|--------------------------------------------|----|----------|-------------------------------------------------------------------------------|----|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------|--------------------------------------------|-----|------------|--------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the edit control (see DIALOGDEFINE).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| <i>Control</i> | Character Expression: A name or number that identifies the edit control (see DIALOGDISPLAY).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the edit control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the edit control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| <i>Width</i>   | Numeric Expression: Width of the edit control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| <i>Height</i>  | Numeric Expression: Height of the edit control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                               |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| <i>Style</i>   | Enumeration: Edit control styles. Type   between enumerations to combine styles. Not all combinations are possible. The values, enumerations, and descriptions are, <table><tr><td>1</td><td>Left!</td><td>Text is left justified.</td></tr><tr><td>2</td><td>Right!</td><td>Text is right justified if Multiline! style is set and WPChars! style is not set.</td></tr><tr><td>4</td><td>Center!</td><td>Text is centered if Multiline! style is set and WPChars! style is not set.</td></tr><tr><td>8</td><td>VScroll!</td><td>Edit control with a vertical scroll bar.</td></tr><tr><td>16</td><td>HScroll!</td><td>Edit control with a horizontal scroll bar.</td></tr><tr><td>32</td><td>WPChars!</td><td>Can display WP Character box (press Ctrl+W). Non-keyboard characters allowed.</td></tr><tr><td>64</td><td>Multiline!</td><td>Multiple lines of input automatically wrap to the next line. Press Enter to cause the default button action (see DIALOGADDPUSHBUTTON). If Enter2Hrtn! style is set in DIALOGDEFINE, press enter to move the insertion point to the next line.</td></tr><tr><td>128</td><td>Uppercase!</td><td>Text automatically converted to uppercase.</td></tr><tr><td>256</td><td>Lowercase!</td><td>Text automatically converted to lowercase.</td></tr></table> | 1                                                                                                                                                                                                                                             | Left! | Text is left justified. | 2 | Right! | Text is right justified if Multiline! style is set and WPChars! style is not set. | 4 | Center! | Text is centered if Multiline! style is set and WPChars! style is not set. | 8 | VScroll! | Edit control with a vertical scroll bar. | 16 | HScroll! | Edit control with a horizontal scroll bar. | 32 | WPChars! | Can display WP Character box (press Ctrl+W). Non-keyboard characters allowed. | 64 | Multiline! | Multiple lines of input automatically wrap to the next line. Press Enter to cause the default button action (see DIALOGADDPUSHBUTTON). If Enter2Hrtn! style is set in DIALOGDEFINE, press enter to move the insertion point to the next line. | 128 | Uppercase! | Text automatically converted to uppercase. | 256 | Lowercase! | Text automatically converted to lowercase. |
| 1              | Left!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Text is left justified.                                                                                                                                                                                                                       |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 2              | Right!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Text is right justified if Multiline! style is set and WPChars! style is not set.                                                                                                                                                             |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 4              | Center!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Text is centered if Multiline! style is set and WPChars! style is not set.                                                                                                                                                                    |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 8              | VScroll!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Edit control with a vertical scroll bar.                                                                                                                                                                                                      |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 16             | HScroll!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Edit control with a horizontal scroll bar.                                                                                                                                                                                                    |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 32             | WPChars!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Can display WP Character box (press Ctrl+W). Non-keyboard characters allowed.                                                                                                                                                                 |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 64             | Multiline!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Multiple lines of input automatically wrap to the next line. Press Enter to cause the default button action (see DIALOGADDPUSHBUTTON). If Enter2Hrtn! style is set in DIALOGDEFINE, press enter to move the insertion point to the next line. |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 128            | Uppercase!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Text automatically converted to uppercase.                                                                                                                                                                                                    |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |
| 256            | Lowercase!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Text automatically converted to lowercase.                                                                                                                                                                                                    |       |                         |   |        |                                                                                   |   |         |                                                                            |   |          |                                          |    |          |                                            |    |          |                                                                               |    |            |                                                                                                                                                                                                                                               |     |            |                                            |     |            |                                            |

|       |             |                                                                                                                  |
|-------|-------------|------------------------------------------------------------------------------------------------------------------|
| 512   | Password!   | Text automatically displayed with asterisk (*) characters. The actual characters are stored by the edit control. |
| 1024  | WordWrap!   | Text wraps if Multiline! style is set.                                                                           |
| 2048  | SoftReturn! | Text contains Soft Return codes [SRt].                                                                           |
| 4096  | Attributes! | Underline (press Ctrl+U) and italics (press Ctrl+I) allowed if WPChars! style is set.                            |
| 8192  | NoTabs!     | Tab codes not exported with output.                                                                              |
| 16384 | NoWPChar!   | Cannot display WP Character dialog box. Non-keyboard characters not allowed.                                     |

### *MacroVar*

Variable: The text entered in the edit box is returned in this variable. Initial text can be assigned to *MacroVar* before the macro compiles.

### *LimitText*

Numeric Expression: The maximum number of characters the edit control accepts.

---

### **Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDFILENAMEBOX, DIALOGADDEDITBOX, and
DIALOGADDVIEWER

VARERRCHK(Off!)
ASSIGN(vText; "Note that DIALOGDEFINE uses Enter2HRtn! style with this edit box control.")
DIALOGDEFINE (1000; 50; 50; 195; 225; OK! |Percent! |Enter2HRtn!; "DIALOGADDCONTROL")
 DIALOGADDDTEXT(1000; -1; 10; 10; 100; 15; Left!; "DIALOGADDFILENAMEBOX")
 DIALOGADDFILENAMEBOX(1000; 100; 10; 20; 170; 15; FilesAndDirs!; vDir; "c:\")
 DIALOGADDDTEXT(1000; -2; 10; 40; 100; 15; Left!; "DIALOGADDEDITBOX")
 DIALOGADDEDITBOX(1000; 101; 10; 50; 170; 50; Left! |Multiline! |WordWrap! |VScroll!; vText;
 1000)
 DIALOGADDDTEXT(1000; -3; 10; 105; 100; 15; Left!; "DIALOGADDVIEWER")
 DIALOGADDVIEWER(1000; 102; 10; 115; 170; 60; "C:\AUTOEXEC.BAT")
DIALOGDISPLAY(1000; 100)
DIALOGDESTROY(1000)

IF(STRLEN(vDir) = 0)
 Msg("You did not enter a path or filename")
ELSE
 MESSAGEBOX(x; "DIALOGADDFILENAMEBOX"; "You entered the following path or path and
 filename: " + vDir)
ENDIF
IF(STRLEN(vText) = 0)
 Msg("You did not enter text in the edit control")
ELSE
 MESSAGEBOX(x; "DIALOGADDEDITBOX"; "You entered the following text in the edit control: " +
 NTOC(0F90Ah) + NTOC(0F90Ah) + vText)
ENDIF

PROCEDURE Msg(x)
 BEEP
 MESSAGEBOX(x; "Oops!"; x; IconExclamation!)
ENDPROC
```

---

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

## DIALOGADDFILENAMEBOX

---

**Purpose**

Adds a filename edit control to a dialog box.

**Syntax**DIALOGADDFILENAMEBOX (*Dialog; Control; Left; Top; Width; Height; Style; MacroVar; DefaultDir; Template*)**Description**

The filename edit control consists of an edit control and a button control. Clicking the button displays the Select Directory or Select File dialog box depending on the control style (see *Style* parameter).

---

**Parameters***Dialog*

Character Expression: The name or number of the dialog box to contain the filename edit control (see DIALOGDEFINE).

*Control*

Character Expression: A unique expression) or number that identifies the filename edit control (see DIALOGDISPLAY).

*Left*

Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the filename edit control.

*Top*

Numeric Expression: Number of dialog units from the top of the dialog box to the top of the filename edit control.

*Width*

Numeric Expression: Width of the filename edit control in dialog units.

*Height*

Numeric Expression: Height of the filename edit control in dialog units.

*Style*

Enumeration: Filename edit control styles. Type | between enumerations to combine styles. The values, enumerations, and descriptions are,

|   |               |                                          |
|---|---------------|------------------------------------------|
| 0 | FilesAndDirs! | Directories and filenames allowed.       |
| 1 | DirOnly!      | Only directories allowed.                |
| 2 | NoValidate!   | No verification that a directory exists. |

*MacroVar*

Variable: The directory or filename entered in the edit box is returned in this variable. An initial name can be assigned to *MacroVar* before the macro compiles.

*DefaultDir*

Character Expression (optional): The default directory when the Select File dialog box is displayed.

*Template*

Character Expression (optional): The filenames to display in the Select File dialog box. For example, "\*".\*" (all files) and "\*.bat" (all batch files).

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDFILENAMEBOX, DIALOGADDEDITBOX, and
DIALOGADDVIEWER

VARERRCHK(Off)
ASSIGN(vText; "Note that DIALOGDEFINE uses Enter2HRtn! style with this edit box control.")
DIALOGDEFINE (1000; 50; 50; 195; 225; OK! | Percent! | Enter2HRtn!; "DIALOGADDCONTROL")
 DIALOGADDTEXT(1000; -1; 10; 10; 100; 15; Left!; "DIALOGADDFILENAMEBOX")
 DIALOGADDFILENAMEBOX(1000; 100; 10; 20; 170; 15; FilesAndDirs!; vDir; "c:\")
 DIALOGADDTEXT(1000; -2; 10; 40; 100; 15; Left!; "DIALOGADDEDITBOX")
 DIALOGADDEDITBOX(1000; 101; 10; 50; 170; 50; Left! | Multiline! | WordWrap! | VScroll!; vText;
 1000)
 DIALOGADDTEXT(1000; -3; 10; 105; 100; 15; Left!; "DIALOGADDVIEWER")
 DIALOGADDVIEWER(1000; 102; 10; 115; 170; 60; "C:\AUTOEXEC.BAT")
DIALOGDISPLAY(1000; 100)
DIALOGDESTROY(1000)

IF(STRLEN(vDir) = 0)
 Msg("You did not enter a path or filename")
ELSE
 MESSAGEBOX(x; "DIALOGADDFILENAMEBOX"; "You entered the following path or path and
 filename: " + vDir)
ENDIF
IF(STRLEN(vText) = 0)
 Msg("You did not enter text in the edit control")
ELSE
 MESSAGEBOX(x; "DIALOGADDEDITBOX"; "You entered the following text in the edit control: " +
 NTOC(0F90Ah) + NTOC(0F90Ah) + vText)
ENDIF

PROCEDURE Msg(x)
 BEEP
 MESSAGEBOX(x; "Oops!"; x; IconExclamation!)
ENDPROC
```

---

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

## DIALOGADDFRAME

---

**Purpose**

Adds a frame to a dialog box.

---

**Syntax**

DIALOGADDFRAME (*Dialog; Control; Left; Top; Width; Height; Style*)

---

**Description**

You can combine frame styles by creating two frames the same size and at the same position. For example, a black frame and a white-filled frame (see *Control* and *Style* parameters). You can also create custom controls by combining frames and hot spots (see DIALOGADDHOTSPOT and DIALOGDISPLAY).

---

## Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------|---|--------|--------------|---|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|---|--------|--------------|---|---------|---------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the frame control (see DIALOGDEFINE).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| <i>Control</i> | Character Expression: A name or number that identifies the frame control (see DIALOGDISPLAY).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the frame control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the frame control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| <i>Width</i>   | Numeric Expression: Width of the frame control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| <i>Height</i>  | Numeric Expression: Height of the frame control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                             |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| <i>Style</i>   | Enumeration: Frame styles. Type   between enumerations to combine styles. Not all combinations are possible. The values, enumerations, and descriptions are,<br><table><tr><td>0</td><td>Frame!</td><td>Frame only.</td></tr><tr><td>0</td><td>Black!</td><td>Black frame.</td></tr><tr><td>1</td><td>Gray!</td><td>Gray frame. Gray is the default DeskTop color of the Windows Default color scheme. To select a different color, choose Colors in the Windows Control Panel.</td></tr><tr><td>2</td><td>White!</td><td>White frame.</td></tr><tr><td>4</td><td>Filled!</td><td>Filled frame.</td></tr></table> | 0                                                                                                                                                           | Frame! | Frame only. | 0 | Black! | Black frame. | 1 | Gray! | Gray frame. Gray is the default DeskTop color of the Windows Default color scheme. To select a different color, choose Colors in the Windows Control Panel. | 2 | White! | White frame. | 4 | Filled! | Filled frame. |
| 0              | Frame!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Frame only.                                                                                                                                                 |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| 0              | Black!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Black frame.                                                                                                                                                |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| 1              | Gray!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Gray frame. Gray is the default DeskTop color of the Windows Default color scheme. To select a different color, choose Colors in the Windows Control Panel. |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| 2              | White!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | White frame.                                                                                                                                                |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |
| 4              | Filled!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Filled frame.                                                                                                                                               |        |             |   |        |              |   |       |                                                                                                                                                             |   |        |              |   |         |               |

### Style combination examples

Frame! | White!

Explanation: White frame displayed.

Filled! | White!

Explanation: White-filled frame displayed.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Use array SortStr to sort up to eight character strings by length
// Calculate width of dialog box from longest string and height from the number of strings
// Demonstrate DIALOGADDFRAME command

CALL(GetNumberAndStrings)
CALL(Dimensions)
CALL(DisplaySortedStrings)

LABEL(GetNumberAndStrings)
 GETNUMBER(vNnbr; "Number of strings to sort (up to eight)"; "FRAME EXAMPLE")
 IF(vNnbr > 8)
 vNnbr := 8
 ENDIF
 DECLARE StrSort[vNnbr]
```



```

FORNEXT(x; 1; vNمبر; 1)
 GETSTRING(StrSort[x]; "String " + x + ";"; "SORT " + vNمبر + " STRINGS"; 50)
ENDFOR
RETURN

LABEL(Dimensions)
vSorted := 1
WHILE(vSorted = 1)
 vSorted := 0
 FOR(x; 1; x < vNمبر; x + 1)
 IF(STRLLEN(StrSort[x]) < STRLEN(StrSort[x + 1]))
 vTemp := StrSort[x]
 StrSort[x] := StrSort[x + 1]
 StrSort[x + 1] := vTemp
 vSorted := 1
 ENDIF
ENDFOR
ENDWHILE
x := 20
y := 20
w := STRLEN(StrSort[1]) + (STRLEN(StrSort[1]) * 4)
h := 13
RETURN

LABEL(DisplaySortedStrings)
DIALOGDEFINE(1000; 50; 50; w + 45; 40 + 30 * vNمبر; Percent!; "FRAME CONTROL")
FORNEXT(r; 1; vNمبر; 1)
 DIALOGADDFRAME(1000; r; x - 5; y - 5 + (r * 30) - 30; w + 10; h + 7; Frame! | Gray!)
 DIALOGADDTTEXT(1000; 0; x; y + (r * 30) - 30; w; h; Center!; TOUPPER(StrSort[r]))
ENDFOR
DIALOGDISPLAY(1000; 1)
DIALOGDESTROY(1000)
QUIT

```

---

**See Also**

♦ DIALOGADDGROUPBOX ♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

## DIALOGADDGROUPBOX

---

**Purpose**

Draws a rectangle around related dialog controls such as buttons.

---

**Syntax**

DIALOGADDGROUPBOX (*Dialog; Control; Left; Top; Width; Height; Title*)

---

**Description**

Grouping related controls helps narrow user options.

---

## Parameters

|                |                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the group box control (see DIALOGDEFINE).           |
| <i>Control</i> | Character Expression: A name or number that identifies the group box control (see DIALOGDISPLAY).                         |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the group box control |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the group box control.            |
| <i>Width</i>   | Numeric Expression: Width of the group box control in dialog units.                                                       |
| <i>Height</i>  | Numeric Expression: Height of the group box control in dialog units.                                                      |
| <i>Title</i>   | Character Expression: The title displayed at the top left corner.                                                         |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate group controls and buttons

VARERRCHK(Off)
DIALOGDEFINE(1000; 50; 50; 200; 140; Percent; "GROUP CONTROLS AND BUTTONS")
 DIALOGADDDGROUPBOX(1000; -1; 10; 10; 75; 55; "Radio Buttons")
 DIALOGADDRADIOBUTTON(1000; 100; 25; 25; 50; 15; "Example 1"; vRadio1)
 DIALOGADDRADIOBUTTON(1000; 101; 25; 45; 50; 15; "Example 2"; vRadio2)
 DIALOGADDVLINE(1000; -2; 125; 15; 55)
 DIALOGADDDPOPUPBUTTON(1000; 102; 135; 35; 50; 13; vPopup)
 DIALOGADDLISTITEM(1000; 102; "Popup 1")
 DIALOGADDLISTITEM(1000; 102; "Popup 2")
 DIALOGADDLISTITEM(1000; 102; "Popup 3")
 DIALOGADDHLINE(1000; -3; 10; 85; 175)
 DIALOGADDPUSHBUTTON(1000; 103; 15; 95; 165; 13; DefaultBtn; "Cancel")
DIALOGDISPLAY(1000; 103)
IF(vRadio1 <> 0)
 vMessage := "You selected radio button 1. "
ELSE
 IF(vRadio2 <> 0)
 vMessage := "You selected radio button 2. "
 ELSE
 vMessage := "You didn't select a radio button. "
 ENDIF
ENDIF
vMessage := vMessage + "You selected " + vPopup + ". "
IF(MacroDialogResult = 103)
 vMessage := vMessage + "You chose Cancel to close the dialog box."
 MESSAGEBOX(x; "MESSAGE BOX"; vMessage; IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "MESSAGE BOX"; "You chose a system command to close the dialog box";
 IconExclamation!)
ENDIF
DIALOGDESTROY(1000)
```

---

**See Also**

- ♦ DIALOGADDFRAME ♦ DIALOGADDHLINE ♦ DIALOGADDVLINE
- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

## DIALOGADDHLINE

---

**Purpose** Adds a horizontal line to a dialog box.

**Syntax** DIALOGADDHLINE (*Dialog; Control; Left; Top; Length*)

---

**Parameters**

- Dialog* Character Expression: The name or number of the dialog box to contain the horizontal line control (see DIALOGDEFINE).
- Control* Character Expression: A name or number that identifies the horizontal line control.
- Left* Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the horizontal line control.
- Top* Numeric Expression: Number of dialog units from the top of the dialog box to the top of the horizontal line control.
- Length* Numeric Expression: Length of the horizontal line control in dialog units.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default: "US")
// Demonstrate group controls and buttons

VARERRCHK(OFF)
DIALOGDEFINE(1000; 50; 50; 200; 140; Percent!; "GROUP CONTROLS AND BUTTONS")
 DIALOGADDDGROUPBOX(1000; -1; 10; 10; 75; 55; "Radio Buttons")
 DIALOGADDRADIOBUTTON(1000; 100; 25; 25; 50; 15; "Example 1"; vRadio1)
 DIALOGADDRADIOBUTTON(1000; 101; 25; 45; 50; 15; "Example 2"; vRadio2)
 DIALOGADDVLINE(1000; -2; 125; 15; 55)
 DIALOGADDDPOPUPBUTTON(1000; 102; 135; 35; 50; 13; vPopup)
 DIALOGADDLISTITEM(1000; 102; "Popup 1")
 DIALOGADDLISTITEM(1000; 102; "Popup 2")
 DIALOGADDLISTITEM(1000; 102; "Popup 3")
 DIALOGADDHLINE(1000; -3; 10; 85; 175)
 DIALOGADDPUSHBUTTON(1000; 103; 15; 95; 165; 13; DefaultBtn!; "Cancel")
DIALOGDISPLAY(1000; 103)
//MESSAGEBOX(x; " "; vradio1)
IF(vRadio1 <> 0)
 vMessage := "You selected radio button 1. "
ELSE
 IF(vRadio2 <> 0)
 vMessage := "You selected radio button 2. "
 ELSE
 vMessage := "You didn't select a radio button. "
ENDIF
```

```

ENDIF
vMessage := vMessage + "You selected " + vPopup + ". "
IF(MacroDialogResult = 103)
 vMessage := vMessage + "You chose Cancel to close the dialog box."
 MESSAGEBOX(x; "MESSAGE BOX"; vMessage; IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "MESSAGE BOX"; "You chose a system command to close the dialog box";
 IconExclamation!)
ENDIF
DIALOGDESTROY(1000)

```

## See Also

- ♦ DIALOGADDDGROUPBOX ♦ DIALOGADDFRAME ♦ DIALOGADDVLINE
- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY

# DIALOGADDHOTSPOT

## Purpose

Creates an invisible area (hot spot) in a dialog box that closes the dialog box when clicked. Other responses are possible when you use DIALOGADDHOTSPOT with a callback function (see DIALOGDISPLAY).

## Syntax

DIALOGADDHOTSPOT (*Dialog; Control; Left; Top; Width; Height; Style*)

## Description

To create a hot spot within a frame (see DIALOGADDFRAME), overlap size and position parameters.

## Parameters

|                |                                                                                                                                                                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the hot spot control (see DIALOGDEFINE).                                                                                                                                                                         |
| <i>Control</i> | Character Expression: A name or number that identifies the hot spot control (see DIALOGDISPLAY). The <i>Control</i> value is returned to the implicit variable MacroDialogResult (see DIALOGADDPUSHBUTTON or DIALOGDEFINE) if the user chooses the hot spot to dismiss the dialog box. |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the hot spot control.                                                                                                                                                              |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the hot spot control.                                                                                                                                                                          |
| <i>Width</i>   | Numeric Expression: Width of the hot spot control in dialog units.                                                                                                                                                                                                                     |
| <i>Height</i>  | Numeric Expression: Height of the hot spot control in dialog units.                                                                                                                                                                                                                    |

Enumeration: Hot spot styles. The values, enumerations, and descriptions are,

- 1 Click! Single click closes dialog box.
- 2 Db!Click! Double click closes dialog box.

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDDHOTSPOT and DIALOGADDICON

VARERRCHK(Off!)
Yes := 6
WM_SYSCOMMAND := 274
DialogControl := 3

DIALOGDEFINE (1000; 50; 50; 115; 85; Percent! | NoFrame!; "DIALOGADDICON")
DIALOGADDDTEXT(1000; -1; 10; 10; 50; 13; Left!; "Click an icon:")
FORNEXT(z; 1; 3; 1)
 DIALOGADDICON (1000; 0; 25; z + 10 + (z * 13); 10; 8; "#" + (z+159); 0)
 DIALOGADDDHOTSPOT(1000; z; 25; z + 10 + (z * 13); 10; 8; Click!)
ENDFOR

DIALOGDISPLAY(1000; 0; Msg)
WHILE(NOT(vStatus = Yes))
ENDWHILE
DIALOGDESTROY (1000)
QUIT

LABEL(Msg)
IF(Msg[5] = WM_SYSCOMMAND)
 vStatus = Yes
ENDIF
SWITCH(Msg[DialogControl])
 CASEOF 1: MESSAGEBOX(vStatus; "DIALOGADDDHOTSPOT"; "Overlap hot spots and
 icons to create custom controls."; IconExclamation!)
 CASEOF 2: MESSAGEBOX(vStatus; "DIALOGADDDHOTSPOT"; "See WordPerfect 6.0 for
 Windows Software Developer's Kit for Shared Code information."; IconInformation!)
 CASEOF 3: MESSAGEBOX(vStatus; "DIALOGADDDHOTSPOT"; "Are you finished with this
 example?"; IconQuestion! | YesNo!)

ENDSWITCH
RETURN
```

---

### See Also

- ♦ DIALOGDEFINE

---

## DIALOGADDICON

### Purpose

Adds an icon to a dialog box.

---

**Syntax**

DIALOGADDICON (*Dialog; Control; Left; Top; Width; Height; IconName; Instance*)

---

**Parameters**

|                       |                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>         | Character Expression: The name or number of the dialog box to contain the icon control (see DIALOGDEFINE).                                               |
| <i>Control</i>        | Character Expression: A name or number that identifies the icon control (see DIALOGDISPLAY).                                                             |
| <i>Left</i>           | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the icon control.                                    |
| <i>Top</i>            | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the icon control.                                                |
| <i>Width</i>          | Numeric Expression: Width of the icon control in dialog units.                                                                                           |
| <i>Height</i>         | Numeric Expression: Height of the icon control in dialog units.                                                                                          |
| <i>IconName</i>       | Character Expression: The name of an icon in a DLL. If an icon is identified by a number, precede the number with a pound sign (#). For example, "#160". |
| <i>ModuleInstance</i> | Numeric Expression: The handle of the DLL that contains the icon. The Shared Code default is 0. DLLLOAD returns the handle in a variable.                |

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDHOTSPOT and DIALOGADDICON

VARERRCHK(O#)
Yes := 6
WM_SYSCOMMAND := 274
DialogControl := 3

DIALOGDEFINE (1000; 50; 50; 115; 85; Percent! | NoFrame!; "DIALOGADDICON")
DIALOGADDTTEXT(1000; -1; 10; 10; 50; 13; Left!; "Click an icon:")
FORNEXT(z; 1; 3; 1)
 DIALOGADDICON (1000; 0; 25; z + 10 + (z * 13); 10; 8; "#" + (z+159); 0)
 DIALOGADDHOTSPOT(1000; z; 25; z + 10 + (z * 13); 10; 8; Click!)
ENDFOR

DIALOGDISPLAY(1000; 0; Msg)
WHILE(NOT(vStatus = Yes))
ENDWHILE
DIALOGDESTROY (1000)
QUIT

LABEL(Msg)
IF(Msg[5] = WM_SYSCOMMAND)
 vStatus = Yes
ENDIF
SWITCH(Msg[DialogControl])
 CASEOF 1: MESSAGEBOX(vStatus; "DIALOGADDHOTSPOT"; "Overlap hot spots and
 icons to create custom controls."; IconExclamation!)
```

```

CASEOF 2: MESSAGEBOX(vStatus: "DIALOGADDDHOTSPOT"; "See WordPerfect 6.0 for
Windows Software Developer's Kit for Shared Code information."; IconInformation!)
CASEOF 3: MESSAGEBOX(vStatus: "DIALOGADDDHOTSPOT"; "Are you finished with this
example?"; IconQuestion! | YesNo!)

```

```

ENDSWITCH
RETURN

```

## See Also

• DIALOGDEFINE • DIALOGDISPLAY • DLLLOAD

# DIALOGADDLISTBOX

## Purpose

Adds a list box control to a dialog box.

## Syntax

DIALOGADDLISTBOX (*Dialog; Control; Left; Top; Width; Height; Style; MacroVar*)

## Description

Displays a list of options to choose from. The option you choose is returned in *MacroVar* (see *MacroVar* parameter). Use DIALOGADDLISTITEM to create list box items.

## Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the list box control (see DIALOGDEFINE).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| <i>Control</i> | Character Expression: A name or number that identifies the list box control (see DIALOGDISPLAY).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the list box control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the list box control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| <i>Width</i>   | Numeric Expression: Width of the list box control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| <i>Height</i>  | Numeric Expression: Height of the list box control in dialog units.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                    |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| <i>Style</i>   | Enumeration: List box control styles. Type   between enumerations to combine styles. The values, enumerations, and descriptions are, <table> <tr> <td>0</td> <td>Unsorted!</td> <td>Items listed in order defined. Type a keyboard character to give the input focus to the first item that begins with that character. A vertical scrollbar automatically appears when needed. See DIALOGADDLISTITEM.</td> </tr> <tr> <td>1</td> <td>Sorted!</td> <td>Default. Items sorted alphabetically. Type a keyboard character to give the input focus to the first item that begins with that character. See DIALOGADDLISTITEM.</td> </tr> </table> | 0                                                                                                                                                                                                                  | Unsorted! | Items listed in order defined. Type a keyboard character to give the input focus to the first item that begins with that character. A vertical scrollbar automatically appears when needed. See DIALOGADDLISTITEM. | 1 | Sorted! | Default. Items sorted alphabetically. Type a keyboard character to give the input focus to the first item that begins with that character. See DIALOGADDLISTITEM. |
| 0              | Unsorted!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Items listed in order defined. Type a keyboard character to give the input focus to the first item that begins with that character. A vertical scrollbar automatically appears when needed. See DIALOGADDLISTITEM. |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |
| 1              | Sorted!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Default. Items sorted alphabetically. Type a keyboard character to give the input focus to the first item that begins with that character. See DIALOGADDLISTITEM.                                                  |           |                                                                                                                                                                                                                    |   |         |                                                                                                                                                                   |

- |    |              |                                                                                                                                                                                                               |
|----|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2  | NameSearch!  | Item search allowed. Begin typing the name of an item to select. An edit control is displayed above the list box, and the input focus is given to the item that most closely matches the characters you type. |
| 4  | MultiColumn! | Items automatically roll into other columns as needed. A horizontal scrollbar automatically appears when needed. Column width is set by Windows.                                                              |
| 32 | WPChars!     | Non-keyboard characters allowed in a list item. See DIALOGADDLISTITEM.                                                                                                                                        |

### Style combination examples

Sort! | NameSearch!

Explanation: Name search allowed on a sorted list box. A vertical scrollbar automatically appears when needed.

Sort! | NameSearch! | MultiColumn!

Explanation: Name search allowed on a sorted list box. Items automatically roll in other columns as needed. A horizontal scrollbar automatically appears when needed.

### MacroVar

Variable: The selected list box item is returned in this variable. An initial item can be assigned to *MacroVar* before the macro compiles.

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDLISTBOX and DIALOGADDLISTITEM

DECLARE Fruit[5]
y := 10
DIALOGDEFINE(1000; 50; 50; 122; 155; OK! | Percent! | NoFrame!; "SORTED LIST BOX")
 FOREACH(vList; {"Apples"; "Oranges"; "Bananas"; "Grapes"; "Pears"})
 DIALOGADDTEXT(1000; 0; 10; y; 30; 15; Left!; vList)
 DIALOGADDHOTSPOT(1000; y; 10; y; 30; 9; Click!)
 FRUIT[y/10] := vList
 y := y + 10
 ENDFOR
DIALOGADDLISTBOX(1000; 103; 50; 10; 60; 50; Sorted!; vChoice)
DIALOGADDTEXT(1000; 0; 10; 75; 102; 35; Left! | RecessBox!; "Click a ""fruit"" to display in the
list box. Click a list box item and choose OK when finished.")
DIALOGDISPLAY(1000; 102; Msg)

Stop := 0
WHILE(Stop = 0)
 ENDWHILE
DIALOGDESTROY(1000)

IF(vChoice = "")
 vMessage := "You didn't select a list box item " + vChoice
ELSE
 vMessage := "You selected " + vChoice
```



```

ENDIF
MESSAGEBOX(x; "LIST BOX CHOICE"; vMessage; IconInformation!)
QUIT

LABEL(Msg)
 IF(Msg[5] = 274)
 Stop := 1
 ENDF
 SWITCH(Msg[3])
 CASEOF 10: DIALOGADDLISTITEM(1000; 103; Fruit[1])
 CASEOF 20: DIALOGADDLISTITEM(1000; 103; Fruit[2])
 CASEOF 30: DIALOGADDLISTITEM(1000; 103; Fruit[3])
 CASEOF 40: DIALOGADDLISTITEM(1000; 103; Fruit[4])
 CASEOF 50: DIALOGADDLISTITEM(1000; 103; Fruit[5])
 CASEOF "OKBtn":
 DIALOGUNDISPLAY(1000; "OKBtn")
 Stop := 1
 ENDSWITCH
RETURN

```

## See Also

- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGADDCOMBOBOX
- ♦ DIALOGADDLISTITEM

---

# DIALOGADDLISTITEM

**Purpose** Creates a list item for DIALOGADDCOMBOBOX, DIALOGADDLISTBOX, or DIALOGADDDPOPUPBUTTON.

**Syntax** DIALOGADDLISTITEM (*Dialog; Control; Item*)

## Parameters

*Dialog* Character Expression: The name or number of the dialog box to contain the list item (see DIALOGDEFINE).

*Control* Character Expression: The name or number of the control to contain the list item.

*Item* Character Expression: A list box item. Use NTOC programming command to include non-keyboard characters in a list item. For example,

```
DIALOGADDLISTITEM(1000; 100; "Sample" + NTOC(5;10))
```

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDLISTBOX and DIALOGADDLISTITEM

DECLARE Fruit[5]
y := 10
DIALOGDEFINE(1000; 50; 50; 122; 155; OK! |Percent! |NoFrame!; "SORTED LIST BOX")
 FOREACH(vList; {"Apples"; "Oranges"; "Bananas"; "Grapes"; "Pears"})

```

```

 DIALOGADDDTEXT(1000; 0; 10; y; 30; 15; Left!; vList)
 DIALOGADDDHOTSPOT(1000; y; 10; y; 30; 9; Click!)
 FRUIT[y/10] := vList
 y := y + 10
 ENDFOR
 DIALOGADDDLSTBOX(1000; 103; 50; 10; 60; 50; Sorted!; vChoice)
 DIALOGADDDTEXT(1000; 0; 10; 75; 102; 35; Left! | RecessBox; "Click a ""fruit"" to display in the
 list box. Click a list box item and choose OK when finished.")
 DIALOGDISPLAY(1000; 102; Msg)

 Stop := 0
 WHILE(Stop = 0)
 ENDWHILE
 DIALOGDESTROY(1000)

 IF(vChoice = "")
 vMessage := "You didn't select a list box item " + vChoice
 ELSE
 vMessage := "You selected " + vChoice
 ENDIF
 MESSAGEBOX(x; "LIST BOX CHOICE"; vMessage; IconInformation!)
 QUIT

 LABEL(Msg)
 IF(Msg[5] = 274)
 Stop := 1
 ENDIF
 SWITCH(Msg[3])
 CASEOF 10: DIALOGADDLSTITEM(1000; 103; Fruit[1])
 CASEOF 20: DIALOGADDLSTITEM(1000; 103; Fruit[2])
 CASEOF 30: DIALOGADDLSTITEM(1000; 103; Fruit[3])
 CASEOF 40: DIALOGADDLSTITEM(1000; 103; Fruit[4])
 CASEOF 50: DIALOGADDLSTITEM(1000; 103; Fruit[5])
 CASEOF "OKBttm":
 DIALOGUNDISPLAY(1000; "OKBttm")
 Stop := 1
 ENDSWITCH
 RETURN

```

---

**See Also**

- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGADDDCOMBOBOX
- ♦ DIALOGADDDLSTBOX ♦ DIALOGADDDPOPUPBUTTON

---

## DIALOGADDDPOPUPBUTTON

**Purpose**

Adds a popup button control to a dialog box.

---

**Syntax** DIALOGADDDPOPUPBUTTON (*Dialog; Control; Left; Top; Width; Height; MacroVar*)

---

**Description** Use DIALOGADDLISTITEM to create items for a popup button control. Unless *MacroVar* is assigned a default item, the item displayed on the popup button is the first item created by DIALOGADDLISTITEM.

---

**Parameters**

*Dialog* Character Expression: The name or number of the dialog box to contain the popup button control (see DIALOGDEFINE).

*Control* Character Expression: A name or number that identifies the popup button control (see DIALOGDISPLAY).

*Left* Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the popup button control.

*Top* Numeric Expression: Number of dialog units from the top of the dialog box to the top of the popup button control.

*Width* Numeric Expression: Width of the popup button control in dialog units.

*Height* Numeric Expression: Height of the popup button control in dialog units.

*MacroVar* Variable: The item displayed on the popup button is returned in this variable. An item can be assigned to *MacroVar* before the macro compiles.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate group controls and buttons

VARERRCHK(Off!)
DIALOGDEFINE(1000; 50; 50; 200; 140; Percent; "GROUP CONTROLS AND BUTTONS")
 DIALOGADDGROUPBOX(1000; -1; 10; 10; 75; 55; "Radio Buttons")
 DIALOGADDRADIOBUTTON(1000; 100; 25; 25; 50; 15; "Example 1"; vRadio1)
 DIALOGADDRADIOBUTTON(1000; 101; 25; 45; 50; 15; "Example 2"; vRadio2)
 DIALOGADDVLINE(1000; -2; 125; 15; 55)
 DIALOGADDDPOPUPBUTTON(1000; 102; 135; 35; 50; 13; vPopup)
 DIALOGADDLISTITEM(1000; 102; "Popup 1")
 DIALOGADDLISTITEM(1000; 102; "Popup 2")
 DIALOGADDLISTITEM(1000; 102; "Popup 3")
 DIALOGADDHLINE(1000; -3; 10; 85; 175)
 DIALOGADDPUSHBUTTON(1000; 103; 15; 95; 165; 13; DefaultBtn!; "Cancel")
DIALOGDISPLAY(1000; 103)

IF(vRadio1 <> 0)
 vMessage := "You selected radio button 1. "
ELSE
 IF(vRadio2 <> 0)
 vMessage := "You selected radio button 2. "
 ELSE
 vMessage := "You didn't select a radio button. "
ENDIF
```

```

ENDIF
vMessage := vMessage + "You selected " + vPopup + ". "
IF(MacroDialogResult = 103)
 vMessage := vMessage + "You chose Cancel to close the dialog box."
 MESSAGEBOX(x; "MESSAGE BOX"; vMessage; IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "MESSAGE BOX"; "You chose a system command to close the dialog box";
 IconExclamation!)
ENDIF
DIALOGDESTROY(1000)

```

---

**See Also**

- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGADDCOMBOBOX
- ♦ DIALOGADDLISTBOX ♦ DIALOGADDLISTITEM

---

## DIALOGADDPUSHBUTTON

---

**Purpose**

Adds a push button to a dialog box.

---

**Syntax**

DIALOGADDPUSHBUTTON (*Dialog; Control; Left; Top; Width; Height; Style; ButtonText*)

---

**Description**

When you choose this button, the *Control* parameter value is returned in the implicit variable MacroDialogResult (see DIALOGDEFINE). If you choose another push button, its *Control* parameter value is returned: 1 for OK, 2 for Cancel, 2 for Close (system menu box), 2 if you double-click the system menu box, 2 if you press Alt+F4, the *Control* parameter value of another user-defined push button, or the *Control* parameter value of a hot spot.

---

**Parameters**

*Dialog*

Character Expression: A name or number that matches the *Dialog* parameter of the dialog box to contain the push button control (see DIALOGDEFINE).

*Control*

Character Expression: A unique string or number that identifies the push button control. DIALOGDISPLAY uses the *Control* parameter to give this control, or another control, the initial input focus.

*Left*

Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the push button control.

*Top*

Numeric Expression: Number of dialog units from the top of the dialog box to the top of the push button control.

*Width*  
*Height*  
*Style*

Numeric Expression: Width of the push button control in dialog units.

Numeric Expression: Height of the push button control in dialog units.

Enumeration: Push button styles. The values, enumerations, and descriptions are,

- |   |                |                                                                                                                                                                                                                                                                                                                                                                                        |
|---|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0 | NonDefaultBtn! | Creates a push button that dismisses the dialog box.                                                                                                                                                                                                                                                                                                                                   |
| 1 | DefaultBtn!    | Creates a default push button, with a heavy black border, that dismisses the dialog box. Press Enter to choose the default button. If Enter2HRtn! style is specified in DIALOGDEFINE, pressing Enter in a multiline edit control produces a hard return rather than the default button action. DIALOGDISPLAY overrides DefaultBtn! if it gives the input focus to another push button. |

*ButtonText*

Character Expression: The text displayed on the push button.

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate group controls and buttons

VARERRCHK(Off)
DIALOGDEFINE(1000; 50; 50; 200; 140; Percent!; "GROUP CONTROLS AND BUTTONS")
 DIALOGADDDGROUPBOX(1000; -1; 10; 10; 75; 55; "Radio Buttons")
 DIALOGADDRADIOBUTTON(1000; 100; 25; 25; 50; 15; "Example 1"; vRadio1)
 DIALOGADDRADIOBUTTON(1000; 101; 25; 45; 50; 15; "Example 2"; vRadio2)
 DIALOGADDVLINE(1000; -2; 125; 15; 55)
 DIALOGADDDPOPUPBUTTON(1000; 102; 135; 35; 50; 13; vPopup)
 DIALOGADDLISTITEM(1000; 102; "Popup 1")
 DIALOGADDLISTITEM(1000; 102; "Popup 2")
 DIALOGADDLISTITEM(1000; 102; "Popup 3")
 DIALOGADDHLINE(1000; -3; 10; 85; 175)
 DIALOGADDPUSHBUTTON(1000; 103; 15; 95; 165; 13; DefaultBtn!; "Cancel")
DIALOGDISPLAY(1000; 103)

IF(vRadio1 <> 0)
 vMessage := "You selected radio button 1. "
ELSE
 IF(vRadio2 <> 0)
 vMessage := "You selected radio button 2. "
 ELSE
 vMessage := "You didn't select a radio button. "
 ENDIF
ENDIF
vMessage := vMessage + "You selected " + vPopup + ". "
IF(MacroDialogResult = 103)
 vMessage := vMessage + "You chose Cancel to close the dialog box."
 MESSAGEBOX(x; "MESSAGE BOX"; vMessage; IconInformation!)
```

```

ELSE
 BEEP
 MESSAGEBOX(x; "MESSAGE BOX"; "You chose a system command to close the dialog box";
 IconExclamation!)
ENDIF
DIALOGDESTROY(1000)

```

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGDESTROY

---

## DIALOGADDRADIOBUTTON

**Purpose**

Adds a radio button to a dialog box.

**Syntax**

DIALOGADDRADIOBUTTON (*Dialog; Control; Left; Top; Width; Height; ButtonText; MacroVar*)

**Description**

Radio buttons represent mutually exclusive choices. Selecting one radio button deselects another. For multiple selections, see DIALOGADDCHECKBOX.

**Parameters**

|                   |                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>     | Character Expression: The name or number of the dialog box to contain the radio button control (see DIALOGDEFINE).                                                                                                                                                                                              |
| <i>Control</i>    | Character Expression: A name or number that identifies the radio button control (see DIALOGDISPLAY).                                                                                                                                                                                                            |
| <i>Left</i>       | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the radio button control.                                                                                                                                                                                   |
| <i>Top</i>        | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the radio button control.                                                                                                                                                                                               |
| <i>Width</i>      | Numeric Expression: Width of the radio button control in dialog units.                                                                                                                                                                                                                                          |
| <i>Height</i>     | Numeric Expression: Height of the radio button control in dialog units.                                                                                                                                                                                                                                         |
| <i>ButtonText</i> | Character Expression: The text displayed to the right of the radio button.                                                                                                                                                                                                                                      |
| <i>MacroVar</i>   | Variable: 1 is returned in this variable if the radio button is selected, and 0 if not. The default value is 0. If <i>MacroVar</i> is assigned a value greater than 0 before the macro compiles, the radio button is automatically selected unless DIALOGDISPLAY gives the input focus to another radio button. |

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate group controls and buttons

```

```

VARERRCHK(Off)
DIALOGDEFINE(1000; 50; 50; 200; 140; Percent; "GROUP CONTROLS AND BUTTONS")
 DIALOGADDDGROUPBOX(1000; -1; 10; 10; 75; 55; "Radio Buttons")
 DIALOGADDRADIOBUTTON(1000; 100; 25; 25; 50; 15; "Example 1"; vRadio1)
 DIALOGADDRADIOBUTTON(1000; 101; 25; 45; 50; 15; "Example 2"; vRadio2)
 DIALOGADDVLINE(1000; -2; 125; 15; 55)
 DIALOGADDDPOPUPBUTTON(1000; 102; 135; 35; 50; 13; vPopup)
 DIALOGADDLISTITEM(1000; 102; "Popup 1")
 DIALOGADDLISTITEM(1000; 102; "Popup 2")
 DIALOGADDLISTITEM(1000; 102; "Popup 3")
 DIALOGADDHLINE(1000; -3; 10; 85; 175)
 DIALOGADDPUSHBUTTON(1000; 103; 15; 95; 165; 13; DefaultBtn!; "Cancel")
DIALOGDISPLAY(1000; 103)

IF(vRadio1 <> 0)
 vMessage := "You selected radio button 1. "
ELSE
 IF(vRadio2 <> 0)
 vMessage := "You selected radio button 2. "
 ELSE
 vMessage := "You didn't select a radio button. "
 ENDIF
ENDIF
vMessage := vMessage + "You selected " + vPopup + ". "
IF(MacroDialogResult = 103)
 vMessage := vMessage + "You chose Cancel to close the dialog box."
 MESSAGEBOX(x; "MESSAGE BOX"; vMessage; IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "MESSAGE BOX"; "You chose a system command to close the dialog box";
 IconExclamation!)
ENDIF
DIALOGDESTROY(1000)

```

---

## See Also

♦ DIALOGADDCHECKBOX ♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

# DIALOGADDSCROLLBAR

## Purpose

Adds a horizontal or vertical scroll bar to a dialog box.

## Syntax

DIALOGADDSCROLLBAR (*Dialog; Control; Left; Top; Width; Height; Style; MacroVar; Minimum; Maximum*)

## Description

A callback function (see DIALOGDISPLAY) can receive Windows WM\_HSCROLL and WM\_VSCROLL messages from DIALOGADDSCROLLBAR. The *wParam* parameter of both messages indicates the mouse location when the scroll bar is clicked, or the thumb position when the thumb is dragged. The values are,

- 0 Right arrow clicked.
- 1 Left arrow clicked.
- 2 Area between the right arrow and thumb clicked.
- 3 Area between the left arrow and thumb clicked.
- 4 Position of thumb after it is dragged, passed as the low-order word of *lParam*.
- 5 The current position of thumb as it is dragged, passed as the low-order word of *lParam*.
- 6 Home key pressed while scroll bar has the input focus.
- 7 End key pressed while scroll bar has the input focus.
- 8 Scroll bar activity has ended.

See WM\_HSCROLL and WM\_VSCROLL in *Programmer's Reference, Volume 3: Messages, Structures, and Macros* (Windows 3.1 documentation).

---

## Parameters

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the scroll bar control (see DIALOGDEFINE).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Control</i> | Character Expression: A name or number that identifies the scroll bar control (see DIALOGDISPLAY).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the scroll bar control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the scroll bar control.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>Width</i>   | Numeric Expression: Width of the scroll bar control in dialog units (0 for vertical scroll bar).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Height</i>  | Numeric Expression: Height of the scroll bar control in dialog units (0 for horizontal scroll bar).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>Style</i>   | Enumeration: Scroll bar thumb position and type styles. Type   between enumerations to combine styles. Not all combinations are possible. The values, enumerations, and descriptions are, <ul style="list-style-type: none"> <li>1 Left! Position the thumb at the left end of a horizontal scroll bar. Initializing <i>MacroVar</i> parameter overrides this parameter.</li> <li>1 Top! Position the thumb at the top end of a vertical scroll bar. Initializing <i>MacroVar</i> parameter overrides this parameter.</li> <li>2 Right! Position the thumb at the right end of a horizontal scroll bar. Initializing <i>MacroVar</i> parameter overrides this parameter.</li> <li>2 Bottom! Position the thumb at the bottom end of a vertical scroll bar. Initializing <i>MacroVar</i> parameter overrides this parameter.</li> </ul> |



- 1 Left! Position the thumb at the left end of a horizontal scroll bar. Initializing *MacroVar* parameter overrides this parameter.
- 8 VScroll! Create a vertical scroll bar. The *Width* parameter does not apply to this style.
- 16 HScroll! Create a horizontal scroll bar. The *Height* parameter does not apply to this style.

**Style combination examples**

Left! | HScroll!

Explanation: Creates a horizontal scroll bar with the thumb at the left end.

Top! | VScroll!

Explanation: Creates a vertical scroll bar with the thumb at the top.

*MacroVar*

Variable: The position of the thumb is returned in this variable. To pre-select a thumb position, initialize *MacroVar* to the desired position.

*Minimum*

Numeric Expression: The minimum thumb position. If the maximum position is 6 and the minimum position is 1, there are six positions on the scroll bar. If the maximum position is 6 and the minimum position is 2, there are five positions on the scroll bar.

*Maximum*

Numeric Expression: The maximum thumb position. See *Minimum* parameter.

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDSCROLLBAR command with callback function

SB_LINEUP := 0
SB_LINEDOWN := 1
WM_SYSCOMMAND := 274
WM_HSCROLL := 276

DIALOGDEFINE(1000; 50; 50; 125; 70; Percent!; "SCROLL BAR")
 DIALOGADDTTEXT(1000; -1; 10; 10; 100; 15; Center! | RecessBox!; "Click the left or right arrow")
 DIALOGADDSCROLLBAR(1000; "Scroll"; 10; 30; 100; 0; Left! | HScroll!; BarPos; 1; 3)

DIALOGDISPLAY(1000; "Scroll"; Msg)
Stop := 1
WHILE(Stop = 1)
ENDWHILE
DIALOGDESTROY(1000)
QUIT

LABEL(Msg)
SWITCH(Msg[5])
 CASEOF WM_SYSCOMMAND: Stop := 0
 CASEOF WM_HSCROLL: CALL(Scroll)
ENDSWITCH
RETURN
```

```

LABEL(Scroll)
SWITCH(Msg[6])
CASEOF SB_LINEUP:
MESSAGEBOX(x; "CALLBACK FUNCTION"; "You clicked the left arrow"; IconInformation!)
CASEOF SB_LINEDOWN:
MESSAGEBOX(x; "CALLBACK FUNCTION"; "You clicked the right arrow"; IconInformation!)
ENDSWITCH
RETURN

```

**See Also**

• DIALOGDEFINE • DIALOGDESTROY • DIALOGDISPLAY

---

## DIALOGADDTXT

---

**Purpose** Displays a line of text in a dialog box.

---

**Syntax** DIALOGADDTXT (*Dialog; Control; Left; Top; Width; Height; Style; Text*)

---

**Parameters**

|                |                                                                                                                                                                                           |                                                                                    |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the text control (see DIALOGDEFINE).                                                                                |                                                                                    |
| <i>Control</i> | Character Expression: A name or number that identifies the text control (see DIALOGDISPLAY).                                                                                              |                                                                                    |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the text control.                                                                     |                                                                                    |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the text control.                                                                                 |                                                                                    |
| <i>Width</i>   | Numeric Expression: Width of the text control in dialog units. This parameter determines how much horizontal space the control occupies, not the text width which is fixed by the system. |                                                                                    |
| <i>Height</i>  | Numeric Expression: Height of the text control in dialog units. This parameter determines how much vertical space the control occupies, not the text height which is fixed by the system. |                                                                                    |
| <i>Style</i>   | Enumeration: Text control styles. Type <code>!</code> between enumerations to combine styles. Not all combinations are possible. The values, enumerations, and descriptions are,          |                                                                                    |
|                | 1                                                                                                                                                                                         | Left! Left justified.                                                              |
|                | 2                                                                                                                                                                                         | Right! Right justified unless WPChars! style is set.                               |
|                | 4                                                                                                                                                                                         | Center! Centered inside <i>Width</i> parameter value unless WPChars! style is set. |
|                | 8                                                                                                                                                                                         | RecessBox! Recessed background.                                                    |

|     |            |                                                                          |
|-----|------------|--------------------------------------------------------------------------|
| 16  | ShadowBox! | Shaded background.                                                       |
| 32  | WPChars!   | Non-keyboard characters allowed. See <i>Text</i> parameter.              |
| 64  | Multiline! | Text wraps if WPChars! is set.                                           |
| 128 | Filename!  | Ellipsis truncate path and filename to length of <i>Width</i> parameter. |

### Style combination examples

Left! | RecessBox!

Explanation: Text is left justified in a recess box.

Center! | ShadowBox!

Explanation: Text is centered in a shadow box.

Character Expression: Text to display on the dialog box. Use NTOC programming command to display a non-keyboard character. For example,

Text: "A bullet character " + NTOC(4; 0)

*Text*

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate UNITSTR and DIALOGADDDTEXT commands

vNmbr := 1
vStatus := 4
DIALOGDEFINE(1000; 50; 50; 100; 120; Percent! | NoTitle! | NoFrame!; "")
 DIALOGADDDTEXT(1000; -1; 10; 15; 75; 15; Left!; "Convert " + vNmbr + " inch to:")
 DIALOGADDCHECKBOX(1000; 101; 15; 30; 100; 13; "Centimeters"; vCB1)
 DIALOGADDCHECKBOX(1000; 102; 15; 45; 100; 13; "Millimeters"; vCB2)
 DIALOGADDCHECKBOX(1000; 103; 15; 60; 100; 13; "Points"; vCB3)
 DIALOGADDCHECKBOX(1000; 104; 15; 75; 100; 13; "WP Units"; vCB4)
 DIALOGADDDTEXT(1000; -2; 10; 100; 100; 15; Left!; "(Press Alt+F4 to cancel)")
DIALOGDISPLAY(1000; -1; Msg)

WHILE(vStatus = 4)
ENDWHILE
DIALOGDESTROY(1000)

LABEL(Msg)
 IF(Msg[5] = 274)
 BEEP
 vStatus := 2
 ENDIF
 DEFAULTUNITS(Inches!)
 SWITCH(Msg[3])
 CASEOF 101: vResult := UNITSTR(vNmbr; Centimeters!)
 Message
 CASEOF 102: vResult := UNITSTR(vNmbr; Millimeters!)
 Message
 CASEOF 103: vResult := UNITSTR(vNmbr; Points!)
 Message
```

```

CASEOF 104: vResult := UNITSTR(vNbr; WPUnts!)
Message
ENDSWITCH
RETURN

LABEL(Message)
DIALOGUNDISPLAY(1000; "CancelBtn")
MESSAGEBOX(vStatus; "UNITSTR COMMAND"; vNbr + " inch is converted to " + vResult;
IconInformation! | RetryCancel!)
IF(vStatus = 4)
 DIALOGDISPLAY(1000; -1; Msg)
ELSE
 BEEP
ENDIF
RETURN

```

---

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGDESTROY

---

## DIALOGADDVIEWER

---

**Purpose** Adds a read-only edit control to a dialog box. The edit control displays a file.

---

**Syntax** DIALOGADDVIEWER (*Dialog; Control; Left; Top; Width; Height; Filename*)

---

**Description** The edit control has horizontal and vertical scroll bars. The horizontal scroll bar contains text that identifies the text file format and indicates whether text wraps.

---

**Parameters**

|                 |                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>   | Character Expression: The name or number of the dialog box to contain the viewer control (see DIALOGDEFINE).            |
| <i>Control</i>  | Character Expression: A name or number that identifies the viewer control (see DIALOGDISPLAY).                          |
| <i>Left</i>     | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the viewer control. |
| <i>Top</i>      | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the viewer control.             |
| <i>Width</i>    | Numeric Expression: Width of the viewer control in dialog units.                                                        |
| <i>Height</i>   | Numeric Expression: Height of the viewer control in dialog units.                                                       |
| <i>Filename</i> | Character Expression: The path and name of a file to view.                                                              |

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate DIALOGADDFILENAMEBOX, DIALOGADDEDITBOX, and
DIALOGADDVIEWER

VARERRCHK(Off)
ASSIGN(vText; "Note that DIALOGDEFINE uses Enter2HRtn! style with this edit box control.")
DIALOGDEFINE (1000; 50; 50; 195; 225; OK! | Percent! | Enter2HRtn!; "DIALOGADDCONTROL")
 DIALOGADDTEXT(1000; -1; 10; 10; 100; 15; Left!; "DIALOGADDFILENAMEBOX")
 DIALOGADDFILENAMEBOX(1000; 100; 10; 20; 170; 15; FilesAndDirs!; vDir; "c:\")
 DIALOGADDTEXT(1000; -2; 10; 40; 100; 15; Left!; "DIALOGADDEDITBOX")
 DIALOGADDEDITBOX(1000; 101; 10; 50; 170; 50; Left! | Multiline! | WordWrap! | VScroll!; vText;
 1000)
 DIALOGADDTEXT(1000; -3; 10; 105; 100; 15; Left!; "DIALOGADDVIEWER")
 DIALOGADDVIEWER(1000; 102; 10; 115; 170; 60; "C:\AUTOEXEC.BAT")
DIALOGDISPLAY(1000; 100)
DIALOGDESTROY(1000)

IF(STRLEN(vDir) = 0)
 Msg("You did not enter a path or filename")
ELSE
 MESSAGEBOX(x; "DIALOGADDFILENAMEBOX"; "You entered the following path or path and
 filename: " + vDir)
ENDIF
IF(STRLEN(vText) = 0)
 Msg("You did not enter text in the edit control")
ELSE
 MESSAGEBOX(x; "DIALOGADDEDITBOX"; "You entered the following text in the edit control: " +
 NTOC(0F90Ah) + NTOC(0F90Ah) + vText)
ENDIF

PROCEDURE Msg(x)
 BEEP
 MESSAGEBOX(x; "Oops!"; x; IconExclamation!)
ENDPROC
```

---

**See Also**

◆ DIALOGDEFINE

---

## DIALOGADDVLINE

---

**Purpose**

Adds a vertical line to a dialog box.

---

**Syntax**

DIALOGADDVLINE (*Dialog; Control; Left; Top; Length*)

---

## Parameters

|                |                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------|
| <i>Dialog</i>  | Character Expression: The name or number of the dialog box to contain the vertical line control (see DIALOGDEFINE).            |
| <i>Control</i> | Character Expression: A name or number that identifies the vertical line control.                                              |
| <i>Left</i>    | Numeric Expression: Number of dialog units from the left side of the dialog box to the left side of the vertical line control. |
| <i>Top</i>     | Numeric Expression: Number of dialog units from the top of the dialog box to the top of the vertical line control.             |
| <i>Length</i>  | Numeric Expression: Length of the vertical line in dialog units.                                                               |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate group controls and buttons

VARERRCHK(Off!)
DIALOGDEFINE(1000; 50; 50; 200; 140; Percent!; "GROUP CONTROLS AND BUTTONS")
 DIALOGADDGROUPBOX(1000; -1; 10; 10; 75; 55; "Radio Buttons")
 DIALOGADDRADIOBUTTON(1000; 100; 25; 25; 50; 15; "Example 1"; vRadio1)
 DIALOGADDRADIOBUTTON(1000; 101; 25; 45; 50; 15; "Example 2"; vRadio2)
 DIALOGADDVLINE(1000; -2; 125; 15; 55)
 DIALOGADDPUPUPBUTTON(1000; 102; 135; 35; 50; 13; vPopup)
 DIALOGADDLISTITEM(1000; 102; "Popup 1")
 DIALOGADDLISTITEM(1000; 102; "Popup 2")
 DIALOGADDLISTITEM(1000; 102; "Popup 3")
 DIALOGADDHLINE(1000; -3; 10; 85; 175)
 DIALOGADDPUSHBUTTON(1000; 103; 15; 95; 165; 13; DefaultBtn!; "Cancel")
DIALOGDISPLAY(1000; 103)

IF(vRadio1 <> 0)
 vMessage := "You selected radio button 1. "
ELSE
 IF(vRadio2 <> 0)
 vMessage := "You selected radio button 2. "
 ELSE
 vMessage := "You didn't select a radio button. "
 ENDIF
ENDIF
vMessage := vMessage + "You selected " + vPopup + ". "
IF(MacroDialogResult = 103)
 vMessage := vMessage + "You chose Cancel to close the dialog box."
 MESSAGEBOX(x; "MESSAGE BOX"; vMessage; IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "MESSAGE BOX"; "You chose a system command to close the dialog box";
 IconExclamation!)
ENDIF
DIALOGDESTROY(1000)
```

---

**See Also**

- ♦ DIALOGADDDGROUPBOX ♦ DIALOGADDDFRAME ♦ DIALOGADDHLINE
- ♦ DIALOGDEFINE ♦ DIALOGDISPLAY

---

# DIALOGDEFINE

---

**Purpose** Creates a dialog box in memory.

**Syntax** DIALOGDEFINE (*Dialog; Left; Top; Width; Height; Style; Caption*)

**Description** Use DIALOGDISPLAY to display a dialog box created by DIALOGDEFINE.

The value of the control that dismisses a dialog box is returned in an implicit variable named MacroDialogResult: 1 for OK, 2 for Cancel, 2 for Close (system menu box), 2 if you double-click the system menu box, 2 if you press Alt+F4, or the value of the *Control* parameter of a user-defined push button or hot spot (see DIALOGADDPUSHBUTTON, DIALOGADDDHOTSPOT).

MacroDialogResult is replaced or destroyed when you close a dialog box or call DIALOGDESTROY. To save its value, assign it to a variable. For example, vResult := MacroDialogResult.

Dialog boxes are positioned and sized in dialog units. A vertical unit equals 1/8 of the font height, and a horizontal unit equals 1/4 of the font width.

---

**Parameters**

*Dialog* Character Expression: A name or number that identifies the dialog box. This name or number is used by DIALOGDISPLAY and by DIALOGADD commands.

*Left* Numeric Expression: Number of dialog units from the left side of the screen to the left side of the dialog box. When used with Percent! (see *Style* parameter), this number is a percentage of the screen width minus the width of the dialog box.

*Top* Numeric Expression: Number of dialog units from the top of the screen to the top of the dialog box. When used with Percent! (see *Style* parameter), this number is a percentage of the screen height minus the height of the dialog box.

*Width* Numeric Expression: Width of the dialog box in dialog units.

*Height* Numeric Expression: Height of the dialog box in dialog units.

*Style* Enumeration: Dialog box styles. Type | between enumerations to combine styles. Not all combinations are possible. The values, enumerations, and descriptions are,

- |   |         |                         |
|---|---------|-------------------------|
| 1 | OK!     | Create an OK button.    |
| 2 | Cancel! | Create a Cancel button. |

|     |             |                                                                                                                                                                                                     |
|-----|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8   | Modeless!   | Dialog box can lose input focus without closing. The WordPerfect command, InhibitInput, must be Off. Default is modal dialog box.                                                                   |
| 16  | Percent!    | Sets <i>Top</i> and <i>Left</i> parameters to a percentage of the screen width or height minus the width or height of the dialog box. Use 50 to display the dialog box in the center of the screen. |
| 32  | NoFrame!    | Create dialog box without a frame.                                                                                                                                                                  |
| 128 | Sizeable!   | Resizing of dialog box allowed.                                                                                                                                                                     |
| 256 | NoTitle!    | Remove the caption (title) bar. Prevents moving the dialog box.                                                                                                                                     |
| 512 | Enter2HRtn! | Press Enter in a multiline edit control (see DIALOGADDEDITBOX) to move the insertion point to the next line. This does not cause the default button action (see DIALOGADDPUSHBUTTON).               |

### Style combination example

OK! †Percent! †NoFrame!

Explanation: Creates a modal dialog box without a frame, and positions it in the center of the screen.

### Caption

Character Expression: The text displayed in the caption (title) bar.

### Example

```
APPLICATION(A1; "WordPerfect"; Default: "US")
// Demonstrate UNITSTR and DIALOGDEFINE commands

vNbr := 1
vStatus := 4
DIALOGDEFINE(1000; 50; 50; 100; 120; Percent! †NoTitle! †NoFrame!; "")
 DIALOGADDTTEXT(1000; -1; 10; 15; 75; 15; Left!; "Convert " + vNbr + " inch to:")
 DIALOGADDCHECKBOX(1000; 101; 15; 30; 100; 13; "Centimeters"; vCB1)
 DIALOGADDCHECKBOX(1000; 102; 15; 45; 100; 13; "Millimeters"; vCB2)
 DIALOGADDCHECKBOX(1000; 103; 15; 60; 100; 13; "Points"; vCB3)
 DIALOGADDCHECKBOX(1000; 104; 15; 75; 100; 13; "WP Units"; vCB4)
 DIALOGADDTTEXT(1000; -2; 10; 100; 100; 15; Left!; "(Press Alt+F4 to cancel)")
DIALOGDISPLAY(1000; -1; Msg)

WHILE(vStatus = 4)
ENDWHILE
DIALOGDESTROY(1000)

LABEL(Msg)
 IF(Msg[5] = 274)
 BEEP
 vStatus := 2
 ENDIF
DEFAULTUNITS(Inches!)
```



```

SWITCH(Msg[3])
 CASEOF 101: vResult := UNITSTR(vNnbr; Centimeters!)
 Message
 CASEOF 102: vResult := UNITSTR(vNnbr; Millimeters!)
 Message
 CASEOF 103: vResult := UNITSTR(vNnbr; Points!)
 Message
 CASEOF 104: vResult := UNITSTR(vNnbr; WPUnts!)
 Message
ENDSWITCH
RETURN

LABEL(Message)
 DIALOGUNDISPLAY(1000; "CancelBttm")
 MESSAGEBOX(vStatus; "UNITSTR COMMAND"; vNnbr + " inch is converted to " + vResult;
 IconInformation! | RetryCancel!)
 IF(vStatus = 4)
 DIALOGDISPLAY(1000; -1; Msg)
 ELSE
 BEEP
 ENDIF
RETURN

```

---

## See Also

♦ DIALOGDISPLAY ♦ DIALOGDESTROY ♦ DIALOGUNDISPLAY

---

# DIALOGDESTROY

---

## Purpose

Destroys (removes from memory) a dialog box.

---

## Syntax

DIALOGDESTROY (*Dialog*)

---

## Parameters

*Dialog* Character Expression: The name or number of a dialog box to destroy (see DIALOGDEFINE).

---

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate UNITSTR and DIALOGDESTROY commands

vNnbr := 1
vStatus := 4
DIALOGDEFINE(1000; 50; 50; 100; 120; Percent! | NoTitle! | NoFrame!; "")
 DIALOGADDTTEXT(1000; -1; 10; 15; 75; 15; Left!; "Convert " + vNnbr + " inch to:")
 DIALOGADDCHECKBOX(1000; 101; 15; 30; 100; 13; "Centimeters"; vCB1)
 DIALOGADDCHECKBOX(1000; 102; 15; 45; 100; 13; "Millimeters"; vCB2)
 DIALOGADDCHECKBOX(1000; 103; 15; 60; 100; 13; "Points"; vCB3)

```

```
DIALOGADDCHECKBOX(1000; 104; 15; 75; 100; 13; "WP Units"; vCB4)
DIALOGADDTTEXT(1000; -2; 10; 100; 100; 15; Left; "(Press Alt+F4 to cancel)")
DIALOGDISPLAY(1000; -1; Msg)
```

```
WHILE(vStatus = 4)
ENDWHILE
DIALOGDESTROY(1000)
```

```
LABEL(Msg)
 IF(Msg[5] = 274)
 BEEP
 vStatus := 2
 ENDIF
DEFAULTUNITS(Inches!)
SWITCH(Msg[3])
 CASEOF 101: vResult := UNITSTR(vNnbr; Centimeters!)
 Message
 CASEOF 102: vResult := UNITSTR(vNnbr; Millimeters!)
 Message
 CASEOF 103: vResult := UNITSTR(vNnbr; Points!)
 Message
 CASEOF 104: vResult := UNITSTR(vNnbr; WPUnits!)
 Message
ENDSWITCH
RETURN
```

```
LABEL(Message)
DIALOGUNDISPLAY(1000; "CancelBtn")
MESSAGEBOX(vStatus; "UNITSTR COMMAND"; vNnbr + " inch is converted to " + vResult;
IconInformation! | RetryCancel!)
IF(vStatus = 4)
 DIALOGDISPLAY(1000; -1; Msg)
ELSE
 BEEP
ENDIF
RETURN
```

---

**See Also**

• DIALOGDEFINE • DIALOGDISPLAY • DIALOGUNDISPLAY

---

## DIALOGDISPLAY

---

**Purpose**

Displays a dialog box, which can have a callback function.

---

**Syntax**

DIALOGDISPLAY (*Dialog; Control; Label*)

---

## Description

A callback function enables a macro to respond immediately and in specific ways to events such as selecting a radio button or check box, without waiting until the dialog box is dismissed. Without a callback function, user input is acted upon only after the dialog box is dismissed and the macro resumes execution.

To create a callback function, use the *Label* parameter of DIALOGDISPLAY, and a loop statement after DIALOGDISPLAY. For example,

```
DIALOGDISPLAY(1000; 1; Messages)
WHILE(x = 0)
ENDWHILE

LABEL(Messages)

... other statements ...

RETURN
```

---

## Parameters

- Dialog* Character Expression: The name or number of the dialog box to display (see DIALOGDEFINE).
- Control* Character Expression: The name or number of the control to receive the initial input focus.
- Label* Label (optional): Identifies a callback function (LABEL to call) when the dialog box becomes active, or the user
- ♦ Chooses Close from the system menu.
  - ♦ Presses Alt+F4.
  - ♦ Double-clicks the system menu box.
  - ♦ Chooses a push button, radio button, check box, hot spot, or scroll bar control.
- The Windows messages are,
- ♦ WM\_ACTIVATE
  - ♦ WM\_COMMAND
  - ♦ WM\_HSCROLL
  - ♦ WM\_SYSCOMMAND
  - ♦ WM\_VSCROLL
- An array is automatically created for callback functions that contains Windows message information. The array is given the same name as the *Label* parameter.
- If *Label* is named DoItNow, array elements are defined as DoItNow[1], DoItNow[2], DoItNow[3], and so forth. The element values are,
- [1] *CallbackType*—Dialog box (type 3).
  - [2] *Dialog*—name (character string) or number that matches the *Dialog* parameter of the dialog box that contains the callback function.
  - [3] *Control*—name (character string) or number that matches the *Control* parameter of the control that causes the callback function to be called. The value is "OKBtn" if OK is pressed; or, "CancelBtn" if Cancel is pressed.

- [4] *DialogHandle*—handle of the dialog box that starts the callback function.
- [5] *Message*—The Windows message received from the dialog box. The values are,
- |     |                                                                                                                    |
|-----|--------------------------------------------------------------------------------------------------------------------|
| 6   | The dialog box becomes active (WM_ACTIVATE).                                                                       |
| 273 | The user chooses a window control, such as a hot spot, push button, or radio button (WM_COMMAND).                  |
| 276 | The user clicks a scroll bar (WM_HSCROLL). See DIALOGADDDSCROLLBAR.                                                |
| 274 | The user chooses Close from the system menu, presses Alt+F4, or double-clicks the system menu box (WM_SYSCOMMAND). |
| 277 | The user clicks a scroll bar (WM_VSCROLL). See DIALOGADDDSCROLLBAR.                                                |
- [6] *wParam*—contains a WORD (16-bit unsigned integer) value that is passed with *Message*. For example, if element 5 equals 273 (WM\_COMMAND), the value of element 6 is 1 if you choose OK, or 2 if you choose Cancel. If element 5 equals 274 (WM\_SYSCOMMAND), the value of element 6 is 61536 if you choose Close from the system menu, press Alt+F4, or double-click the system menu box.
- [7] *lParam*—contains a DWORD (32-bit unsigned integer) value that is passed with *Message*. This value contains a high-order value and a low-order value, which you calculate with bitwise operators. The low-order value equals  $x := x \& 65535$  (mask hi-order value). The high-order value equals  $x := x \gg 16$  (shift right 16 bits).

Elements 6 (*wParam*) and 7 (*lParam*) have different meanings for every Windows message. See specified messages in *Programmer's Reference, Volume 3: Messages, Structures, and Macros* (Windows 3.1 documentation).

## Example

```
APPLICATION(WP; "WordPerfect"; Default; "US")

HRt = NTOC(0F90Ah)
HSp = " "
vCombo = "Combo item 1"
vList = "List item 1"
WM_COMMAND = 273
WM_SYSCOMMAND = 274

DIALOGDEFINE(1000; 50; 50; 160; 125; Percent! | NoFrame! | Cancel!; "WM_COMMAND")
DIALOGADDLISTBOX(1000; 100; 10; 10; 60; 30; Sorted; vList)
DIALOGADDLISTITEM(1000; 100; "List item 1")
DIALOGADDLISTITEM(1000; 100; "List item 2")
DIALOGADDLISTITEM(1000; 100; "List item 3")
DIALOGADDCOMBOBOX(1000; 101; 10; 50; 60; 40; Droplist!; vCombo; 100)
DIALOGADDLISTITEM(1000; 101; "Combo item 1")
DIALOGADDLISTITEM(1000; 101; "Combo item 2")
DIALOGADDLISTITEM(1000; 101; "Combo item 3")
DIALOGADDPUPUPBUTTON(1000; 102; 10; 80; 60; 15; vPopup)
DIALOGADDLISTITEM(1000; 102; "Popup item 1")
DIALOGADDLISTITEM(1000; 102; "Popup item 2")
DIALOGADDLISTITEM(1000; 102; "Popup item 3")
```

```

DIALOGADDPUSHBUTTON(1000; 103; 100; 10; 50; 15; NonDefaultBtn!; "Push Button")
DIALOGADDRADIOBUTTON(1000; 104; 100; 35; 50; 15; "Radio"; vRadio)
DIALOGADDCHECKBOX(1000; 105; 100; 60; 50; 15; "Check box"; vCheck)
DIALOGHANDLE(hButton; 1000; 103)
DIALOGHANDLE(hRadio; 1000; 104)
DIALOGHANDLE(hCheck; 1000; 105)
DIALOGDISPLAY(1000; 1; Msg)
x := 0
While(x = 0)
Endwhile
DIALOGDESTROY(1000)

LABEL(MSG)
IF((Msg[5] = WM_SYSCOMMAND) OR (Msg[3] = "CancelBtn"))
 BEEP
 x = 1
 RETURN
ENDIF
IF(Msg[5] = WM_COMMAND)
 IParam = Msg[7]
 DIALOGUNDISPLAY(1000; 1)
 MsgBox
 DIALOGDISPLAY(1000; 1; Msg)
ENDIF
RETURN

LABEL(MsgBox)
SWITCH(LOWRD(IParam))
 CASFOF hButton: vHandle = hButton
 CASEOF hRadio: vHandle = hRadio
 CASEOF hCheck: vHandle = hCheck
 DEFAULT: vHandle = "N/A"
ENDSWITCH
Lo = "Low-word: " + LOWRD(IParam) + HRt + HRt
Hi = "High-word: " + HIWRD(IParam) + HRt + HRt
Handle = "Control handle: " + vHandle + HRt + HRt
ListItems = "Selected list items: " + HRt + HRt + HSp + vPopup + HRt + HSp + vList + HRt + HSp +
vCombo + HRt
MESSAGEBOX(vResult; "WM_COMMAND"; Hi + Lo + Handle + ListItems)
RETURN

FUNCTION LOWRD(lo)
 lo = lo & 65535
 Return(lo)
ENDFUNC

FUNCTION HIWRD(hi)
 hi = hi >> 16
 RETURN(hi)
ENDFUNC

```

---

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDESTROY ♦ DIALOGUNDISPLAY

---

## DIALOGHANDLE

---

**Purpose**

Returns the handle of a dialog box or dialog box control.

---

**Syntax**DIALOGHANDLE (*MacroVar*; *Dialog*; *Control*)

---

**Returns**The handle of a dialog box or dialog box control (see *MacroVar* parameter).

---

**Parameters***MacroVar*Variable: The handle of a dialog box or dialog box control is returned in this variable (see *Dialog* and *Control* parameters).*Dialog*

Character Expression: The name or number of a dialog box (see DIALOGDEFINE).

*Control*Character Expression (optional): The name or number of a dialog box control (see *Control* parameter of any dialog control). If you omit this parameter, the handle of the dialog box is returned in *MacroVar*.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Returns the handle of five push button controls
// Demonstrate DIALOGHANDLE command

VARERRCHK(Off)
DIALOGDEFINE(1000; 50; 50; 100; 190; Percent!; "DIALOGHANDLE")
FORNEXT(x; 1; 5; 1)
 DIALOGADDPUSHBUTTON(1000; x; 10; x + (x * 20) - 10; 75; 15; NonDefaultBtn!; "Button "
 + x)
ENDFOR
DIALOGADDTEXT(1000; -1; 10; 120; 75; 40; RecessBox! | Left!; "Press Enter, or choose Close from
the system menu box.")
DIALOGDISPLAY(1000; 1)
vResult := ""
FOREACH(vControl; {"1"; "2"; "3"; "4"; "5"})
 vResult := vResult + "Button " + vControl + ": "
 DIALOGHANDLE(vHandles; 1000; vControl)
 vResult := vResult + vHandles + ". "
ENDFOR
MESSAGEBOX(x; "CONTROL HANDLES"; vResult)
DIALOGDESTROY(1000)
```

---

**See Also**

♦ DIALOGADDCONTROL ♦ DIALOGDEFINE

---

# DIALOGUNDISPLAY

---

|                    |                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Hides a dialog box.                                                                                                                              |
| <b>Syntax</b>      | DIALOGUNDISPLAY ( <i>Dialog</i> ; <i>Control</i> )                                                                                               |
| <b>Description</b> | DIALOGUNDISPLAY only hides a dialog box. Use DIALOGDESTROY to remove a dialog box from memory. Use DIALOGDISPLAY to display a hidden dialog box. |

---

## Parameters

*Dialog* Character Expression: The name or number of the dialog box to hide (see DIALOGDEFINE).

*Control* Character Expression: OKBtn or 1 for OK, CancelBtn or 2 for Cancel, or the control value of a user-defined push button or hot spot. CancelBtn cancels value assignments to control return variables. For example, an edit control does not return to *MacroVar* the text entered in the edit box (see DIALOGADDEDITBOX). The value of *Control* is returned in MacroDialogResult (see DIALOGDEFINE) as OKBtn, CancelBtn, or the control value of another push button or hot spot.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate UNITSTR and DIALOGUNDISPLAY commands

vNbr := 1
vStatus := 4
DIALOGDEFINE(1000; 50; 50; 100; 120; Percent! | NoTitle! | NoFrame!; "")
 DIALOGADDTXT(1000; -1; 10; 15; 75; 15; Left; "Convert " + vNbr + " inch to:")
 DIALOGADDCHECKBOX(1000; 101; 15; 30; 100; 13; "Centimeters"; vCB1)
 DIALOGADDCHECKBOX(1000; 102; 15; 45; 100; 13; "Millimeters"; vCB2)
 DIALOGADDCHECKBOX(1000; 103; 15; 60; 100; 13; "Points"; vCB3)
 DIALOGADDCHECKBOX(1000; 104; 15; 75; 100; 13; "WP Units"; vCB4)
 DIALOGADDTXT(1000; -2; 10; 100; 100; 15; Left; "(Press Alt+F4 to cancel)")
DIALOGUNDISPLAY(1000; -1; Msg)

WHILE(vStatus = 4)
ENDWHILE
DIALOGDESTROY(1000)

LABEL(Msg)
 IF(Msg[5] = 274)
 BEEP
 vStatus := 2
 ENDIF
 DEFAULTUNITS(Inches!)
 SWITCH(Msg[3])
 CASEOF 101: vResult := UNITSTR(vNbr; Centimeters!)
 Message
 CASEOF 102: vResult := UNITSTR(vNbr; Millimeters!)
 Message
 CASEOF 103: vResult := UNITSTR(vNbr; Points!)
```

```
Message
CASEOF 104: vResult := UNITSTR(vNnbr; WPUints!)
Message
ENDSWITCH
RETURN
```

```
LABEL(Message)
DIALOGUNDISPLAY(1000; "CancelBtn")
MESSAGEBOX(vStatus; "UNITSTR COMMAND"; vNnbr + " inch is converted to " + vResult;
IconInformation! | RetryCancel!)
IF(vStatus = 4)
 DIALOGDISPLAY(1000; -1; Msg)
ELSE
 BEEP
ENDIF
RETURN
```

---

**See Also**

♦ DIALOGDEFINE ♦ DIALOGDISPLAY ♦ DIALOGDESTROY

---

## DISCARD

---

**Purpose**

Removes LOCAL, GLOBAL, and PERSIST variables from memory, in that order.

---

**Syntax**

DISCARD (*VariableName*; *VariableName*; ... *VariableName*)

---

**Description**

DISCARD does not specify a variable table. If a variable by the same name exists in all three tables, call DISCARD three times.

```
WHILE(EXISTS(VariableName))
DISCARD(VariableName)
ENDWHILE
```

---

**Parameters**

*VariableName*

Variable: Begins with a letter and can include any other combination of letters or numbers. Separate multiple variables with a semicolon.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Declare and test variable types
// Demonstrate DISCARD, BREAK, EXIST, LOCAL, GLOBAL, and PERSIST commands

DISCARD var5; var1; var7
LOCAL var2; var9; var4
GLOBAL var3; var10; var6
PERSIST var5; var1; var7
ASSIGN(var8; "")
```



```

FOR(x; 1; x < 11; x + 1)
 ASSIGN(INDIRECT("var" + x); x)
 MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion! | YesNo!)
 IF(vStatus = 7)
 BREAK
 ENDIF
ENDFOR

FOR(z; 1; z < (x + 1); z + 1)
 SWITCH(EXISTS(INDIRECT("var" + z)))
 CASEOF 1: Msg("Local"; z)
 CASEOF 2: Msg("Global"; z)
 CASEOF 3: Msg("Persist"; z)
 ENDSWITCH
ENDFOR

FUNCTION Msg(w; y)
 MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w)
ENDFUNC

```

---

## See Also

♦ EXISTS ♦ GLOBAL ♦ LOCAL ♦ PERSIST ♦ PERSISTALL

---

# DLLCALL

## Purpose

Calls a function in a Dynamic Link Library (DLL).

**IMPORTANT:** Do not use this command unless you are familiar with Windows programming. Incorrect DLL calls can damage files and/or reset your computer.

## Syntax

*DLLCALL (ModuleInstance; FunctionName; ReturnVariable; Return Type; {Parameter; Parameter; ...})*

## Parameters

|                       |                                                                                  |
|-----------------------|----------------------------------------------------------------------------------|
| <i>ModuleInstance</i> | Numeric Expression: A DLL handle. DLLLOAD returns the handle in a variable.      |
| <i>FunctionName</i>   | Character Expression: A function name. See DLL documentation for function names. |
| <i>ReturnVariable</i> | Variable: The DLL function may return a value in this variable.                  |
| <i>Return Type</i>    | Enumeration: The data type returned in <i>ReturnVariable</i> .                   |

| Data Type  | Description                |
|------------|----------------------------|
| WString    | WordPerfect 6.0 wordstring |
| AnsiString | Windows type LPSTR         |
| OemString  | Windows type LPSTR         |
| String     | Windows type LPSTR         |
| Bool       | Windows type BOOL          |

|         |                         |
|---------|-------------------------|
| DWord   | Windows type DWORD      |
| Integer | Windows type LONG       |
| Word    | Windows type WORD       |
| Real    | Microsoft C type double |
| Void    | Nothing                 |

IMPORTANT: *Return Type* must match the data type returned to *Return Variable* (see DLL function documentation for return types). If it does not, an Unrecoverable Application Error or system crash may result.

### Parameter

Any: Parameter information passed to a DLL function, enclosed in braces and separated by a semicolon. The type of information depends on the function being called (see DLL function documentation for parameter values to pass).

| <b>Data Types</b>                    | <b>Expected Data Types</b>                                   |
|--------------------------------------|--------------------------------------------------------------|
| <i>Integer</i>                       | LONG or DWORD                                                |
| <i>String</i>                        | FAR pointer to an ANSI string                                |
| <i>Boolean</i>                       | BOOL                                                         |
| <i>Real</i>                          | Microsoft c-type double                                      |
| ANSISTRING( <i>String</i> )          | FAR pointer to an ANSI string                                |
| OEMSTRING( <i>String</i> )           | FAR pointer to an OEM string                                 |
| WString( <i>String</i> )             | FAR pointer to a WP 6.0 word string                          |
| LOWORD( <i>Integer</i> )             | int, short, or WORD (low order bits passed to DLL function)  |
| HIWORD( <i>Integer</i> )             | int, short, or WORD (high order bits passed to DLL function) |
| ADDRESS( <i>Integer</i> )            | LPDWORD                                                      |
| ADDRESS(WORD( <i>Integer</i> ))      |                                                              |
| ADDRESS(DWORD( <i>Integer</i> ))     |                                                              |
| ADDRESS( <i>Boolean</i> )            | BOOL FAR*                                                    |
| ADDRESS(BOOLEAN( <i>Boolean</i> ))   | BOOL FAR*                                                    |
| ADDRESS( <i>String</i> )             | FAR pointer to an ANSI string                                |
| ADDRESS(ANSISTRING( <i>String</i> )) | FAR pointer to an ANSI string                                |
| ADDRESS(OEMSTRING( <i>String</i> ))  | FAR pointer to OEM string                                    |
| ADDRESS(WPSTRING( <i>String</i> ))   | FAR pointer to a WP 6.0 word string.                         |
| ADDRESS( <i>Real</i> )               | FAR pointer to a MicroSoft C-type double                     |
| ADDRESS(REAL( <i>Real</i> ))         | FAR pointer to a MicroSoft C-type double                     |

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Change current working directory
// Demonstrate DLLLOAD, DLLCALL, and DLLFREE
// Assign hard return code to a variable, insert in prompt string
```

// For information about WfsSetCurDir and WfsGetCurDir, see *WordPerfect 6.0 for Windows Software Developer's Kit*

```
DLLLOAD(hInstance; "shwin20.dll")
DLLCALL(hInstance; "WfsGetCurDir"; vMsgResult: WORD; {ADDRESS (ANSISTRING(CurDir))})

ASSIGN(HdReturn; NTOC(0F90Ah)) // hard return code
LABEL(Start)
 GETSTRING(vNewDir; "The current working directory is " + CurDir + HdReturn + HdReturn +
 "Enter a new working directory: "; "DLL EXAMPLE.")
 IF(vNewDir = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering a path"; IconStop!)
 GO(Start)
 ENDIF
 DLLCALL(hInstance; "WfsSetCurDir"; vMsgResult: WORD; {vNewDir})
 DLLCALL(hInstance; "WfsGetCurDir"; vMsgResult: WORD; {ADDRESS (ANSISTRING(CurDir))})
 MESSAGEBOX(vAns; "DLL EXAMPLE"; "The new working directory is " + CurDir; IconInformation!
 | RetryCancel!)
 IF(vAns = 2)
 BEEP
 QUIT
 ELSE
 GO(Start)
 ENDIF

 DLLFREE(hInstance)
 QUIT
```

---

## See Also

♦ DLLFREE ♦ DLLLOAD

---

# DLLFREE

---

## Purpose

Removes a dynamic link library (DLL) from memory.

**IMPORTANT:** If you use DLL commands incorrectly, they can damage files and/or reset your computer.

---

## Syntax

DLLFREE (*ModuleInstance*)

---

## Parameters

*ModuleInstance*

Numeric Expression: A DLL handle. DLLLOAD returns the handle in a variable.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Change current working directory
// Demonstrate DLLLOAD, DLLCALL, and DLLFREE
// Assign hard return code to a variable, insert in prompt string
```

```
// For information about WfsSetCurDir and WfsGetCurDir, see WordPerfect 6.0 for Windows Software Developer's Kit

DLLLOAD(hInstance; "shwin20.dll")
DLLCALL(hInstance; "WfsGetCurDir"; vMsgResult: WORD; {ADDRESS (ANSISTRING(CurDir))})

ASSIGN(HdReturn; NTOC(0F90Ah)) // hard return code
LABEL(Start)
 GETSTRING(vNewDir; "The current working directory is " + CurDir + HdReturn + HdReturn +
 "Enter a new working directory: "; "DLL EXAMPLE")
 IF(vNewDir = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering a path"; IconStop!)
 GO(Start)
 ENDIF
 DLLCALL(hInstance; "WfsSetCurDir"; vMsgResult: WORD; vNewDir)
 DLLCALL(hInstance; "WfsGetCurDir"; vMsgResult: WORD; {ADDRESS (ANSISTRING(CurDir))})
 MESSAGEBOX(vAns; "DLL EXAMPLE"; "The new working directory is " + CurDir; IconInformation!
 | RetryCancel!)
 IF(vAns = 2)
 BEEP
 QUIT
 ELSE
 GO(Start)
 ENDIF

 DLLFREE(hInstance)
 QUIT
```

---

## See Also

♦ DLLCALL ♦ DLLLOAD

---

# DLLLOAD

---

## Purpose

Loads a dynamic link library (DLL) into memory.

**IMPORTANT:** If you use DLL commands incorrectly, they can damage files and/or reset your computer.

---

## Syntax

DLLLOAD (*ModuleInstance*; *Filename*)

---

## Description

You should respond to error values with code similar to the following (see *ModuleInstance* parameter):

```
DLLLOAD(vResult; Filename)
SWITCH(vResult)
CASEOF 0: ...statement block...
CASEOF 2: ...statement block...
ENDSWITCH
```

---

## Parameters

### *ModuleInstance*

Numeric Expression: A DLL handle if the DLL successfully loads, or an error value less than 32 if not. Partial list of error values:

- 0 Not enough system memory.
- 2 File not found.
- 3 Path not found.
- 10 Incorrect Windows version.
- 20 DLL is invalid.

See LoadLibrary function in *Programmer's Reference, Volume 2: Functions* (Windows 3.1 documentation) for the complete list of error values.

### *Filename*

Character Expression: A DLL path and filename.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Change current working directory
// Demonstrate DLLLOAD, DLLCALL, and DLLFREE
// Assign hard return code to a variable, insert in prompt string
// For information about WfsSetCurDir and WfsGetCurDir, see WordPerfect 6.0 for Windows Software Developer's Kit

DLLLOAD(hInstance; "shwin20.dll")
DLLCALL(hInstance; "WfsGetCurDir"; vMsgResult: WORD; {ADDRESS (ANSISTRING(CurDir))})

ASSIGN(HdReturn; NTOC(0F90Ah)) // hard return code
LABEL(Start)
 GETSTRING(vNewDir; "The current working directory is " + CurDir + HdReturn + HdReturn +
 "Enter a new working directory: "; "DLL EXAMPLE")
 IF(vNewDir = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering a path"; IconStop!)
 GO(Start)
 ENDIF
 DLLCALL(hInstance; "WfsSetCurDir"; vMsgResult: WORD; {vNewDir})
 DLLCALL(hInstance; "WfsGetCurDir"; vMsgResult: WORD; {ADDRESS (ANSISTRING(CurDir))})
 MESSAGEBOX(vAns; "DLL EXAMPLE"; "The new working directory is " + CurDir; IconInformation!
 | RetryCancel!)
 IF(vAns = 2)
 BEEP
 QUIT
 ELSE
 GO(Start)
 ENDIF

 DLLFREE(hInstance)
 QUIT
```

---

**See Also** ♦ DLLFREE ♦ DLLCALL

---

## ENDAPP

---

**Purpose** Identifies, for the compiler, an application that is no longer used in a macro (see APPLICATION).

---

**Syntax** ENDAPP (*ProductPrefix*)

---

**Description** ENDAPP is a non-executable statement that can occur anywhere in a macro. After ENDAPP, a product command to the application specified by ENDAPP creates a compile-time syntax error.

---

### Parameters

*ProductPrefix* Character Expression: A unique identifier that matches the *ProductPrefix* parameter of an APPLICATION statement.

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default: "US")
// Compiling this macro produces a compile-time syntax error
// To compile, make the second AboutDlg a comment (//)
// Demonstrate ENDAPP command

AboutDlg

ENDAPP(A1)

AboutDlg
```

---

**See Also** ♦ APPLICATION ♦ NEWDEFAULT

---

## ERROR

---

**Purpose** Determines how a macro responds to an Error condition.

---

**Syntax** ERROR (*State*)

---

**Description** Create an Error condition with ASSERT(ErrorCondition!).

---

---

## Parameters

### State

Enumeration: Specifies the Error state. The default is ERROR(On!). The enumerations are,

- Off! Overrides an Error condition.
- On! Stops a macro unless preceded by ONERROR, which directs macro execution to a specified LABEL.

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// LABEL(ErrorMsg) executes with ERROR(On!)
// Demonstrate ERROR command

ONERROR(ErrorMsg)

REPEAT
 MENU(vChoice; Digit;;; {"ERROR(Off!); "ERROR(On!); "Quit"})
 CASE CALL(vChoice; {1; OffMsg; 2; OnMsg; 3; QuitMacro}; QuitMacro)
UNTIL(vChoice = 3)

LABEL(OffMsg)
 ERROR(Off!)
 ASSERT(ErrorCondition!) // ignore Error condition
 MESSAGEBOX(x; "ERROR(Off!); "Error condition ignored"; IconInformation!)
 RETURN

LABEL(OnMsg)
 ERROR(On!)
 ASSERT(ErrorCondition!) // LABEL(ErrorMsg) executed
 RETURN

LABEL(ErrorMsg)
 BEEP
 MESSAGEBOX(x; "ERROR CONDITION"; "An error condition causes LABEL(ErrorMsg) to
execute"; IconInformation!)
 RETURN

LABEL(QuitMacro)
 QUIT
```

---

### See Also

♦ ASSERT ♦ CANCEL ♦ LABEL ♦ NOTFOUND ♦ ONERROR

---

## EXISTS

### Purpose

Determines if a variable exists.

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>       | ReturnValue := EXISTS ( <i>VariableName</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Returns</b>      | An integer. The values are, <ul style="list-style-type: none"> <li>0 Does not exist</li> <li>1 Local variable</li> <li>2 Global variable</li> <li>3 Persist variable</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Description</b>  | <p>To exist, a variable must be declared and initialized. Given that variable w is declared and not initialized, and that LOCAL x is initialized to 5, GLOBAL y to 10, and PERSIST z to 15, the following statements are valid:</p> <pre>vNoInit := EXISTS(w) // vNoInit = 0</pre> <p>Result: vNoInit equals 0</p> <pre>vLocal := EXISTS(x) // vLocal = 1</pre> <p>Result: vLocal = 1</p> <pre>vGlobal := EXISTS(y) // vGlobal = 2</pre> <p>Result: vGlobal equals 2</p> <pre>vPersist := EXISTS(z) // vPersist = 3</pre> <p>Result: vPersist equals 3</p> <p><b>Examples in shorthand notation</b></p> <pre>IF(EXISTS(vLocal)) BEEP ENDIF</pre> <p>Explanation: Computer beeps because EXISTS returns a value greater than 0.</p> <pre>IF(EXISTS(vLocal) = 3) BEEP ENDIF</pre> <p>Explanation: The computer does not beep because EXISTS returns 1. (vLocal is a Local variable, not a Persist variable).</p> |
| <b>Parameters</b>   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>VariableName</i> | Variable: A variable declared as LOCAL, GLOBAL, or PERSIST. Variables begin with a letter and can include any other combination of letters or numbers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Example</b>      | <pre>APPLICATION(A1; "WordPerfect"; Default; "US") // Declare and test variable types // Demonstrate BREAK, EXISTS, LOCAL, GLOBAL, and PERSIST commands  DISCARD var5; var1; var7 LOCAL var2; var9; var4</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



```

GLOBAL var3; var10; var6
PERSIST var5; var1; var7
ASSIGN(var8; "")

FOR(x; 1; x < 11; x + 1)
 ASSIGN(INDIRECT("var" + x); x)
 MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion! | YesNo!)
 IF(vStatus = 7)
 BREAK
 ENDIF
ENDFOR

FOR(z; 1; z < (x + 1); z + 1)
 SWITCH(EXISTS(INDIRECT("var" + z)))
 CASEOF 1: Msg("Local"; z)
 CASEOF 2: Msg("Global"; z)
 CASEOF 3: Msg("Persist"; z)
 ENDSWITCH
ENDFOR

FUNCTION Msg(w; y)
 MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w)
ENDFUNC

```

---

## See Also

♦ DISCARD ♦ LOCAL ♦ GLOBAL ♦ PERSIST ♦ PERSISTALL

---

# FOR-ENDFOR

---

**Purpose** A loop statement that executes a specified number of times (see Loop Statements).

---

**Syntax** The general form of a FOR statement is,

```

FOR (ControlVariable; InitialValue; TerminateExp; IncrementExp)
...statement block...
ENDFOR

```

For example, the following statement initializes vTest to one, repeats the statement block while vTest is less than ten, and increments vTest by two after each loop.

```

FOR(vTest; 1; vTest < 10; vTest + 2)
...statement block...
ENDFOR

```

---

**Description** The loop executes only if *TerminateExp* is true, and continues to execute until *ControlVariable*'s value makes *TerminateExp* false.

ENDFOR closes a FOR statement.

---

## Parameters

|                        |                                                                             |
|------------------------|-----------------------------------------------------------------------------|
| <i>ControlVariable</i> | Variable: Control variable.                                                 |
| <i>InitialValue</i>    | Any: Value assigned to <i>ControlVariable</i> at the start of a loop.       |
| <i>TerminateExp</i>    | Relational Expression: The loop executes while <i>TerminateExp</i> is true. |
| <i>IncrementExp</i>    | Any: The value to increment <i>ControlVariable</i> after each loop.         |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Declare and test variable types
// Demonstrate FOR, BREAK, EXIST, LOCAL, GLOBAL, and PERSIST commands

DISCARD var5; var1; var7
LOCAL var2; var9; var4
GLOBAL var3; var10; var6
PERSIST var5; var1; var7
ASSIGN(var8; "")

FOR(x; 1; x < 11; x + 1)
 ASSIGN(INDIRECT("var" + x); x)
 MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion! ! YesNo!)
 IF(vStatus = 7)
 BREAK
 ENDIF
ENDFOR

FOR(z; 1; z < (x + 1); z + 1)
 SWITCH(EXISTS(INDIRECT("var" + z)))
 CASEOF 1: Msg("Local"; z)
 CASEOF 2: Msg("Global"; z)
 CASEOF 3: Msg("Persist"; z)
 ENDSWITCH
ENDFOR

FUNCTION Msg(w; y)
 MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w)
ENDFUNC
```

---

## See Also

♦ FORNEXT ♦ FOREACH ♦ REPEAT ♦ WHILE

---

# FOREACH-ENDFOR

---

## Purpose

A loop statement that executes a number of times equal to the number of specified expressions. See *Loop Statements* in *Macro Control Statements*.

---

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | <p>The general form of a FOREACH statement is,</p> <pre>FOREACH (<i>ControlVariable</i>; {<i>ValueList</i>; ...}) ...<i>statement block</i>... ENDFOR</pre> <p>For example, the following statement repeats three times. The variable vTest is initialized to "Apples" before the first loop, to "Oranges" before the second, and to "Bananas" before the third.</p> <pre>FOREACH(vTest; {"Apples"; "Oranges"; "Bananas"}) ... <i>statement block</i> ... ENDFOR</pre> |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                    |                                                                                                                                                                                                                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Description</b> | <p><i>ControlVariable</i> is assigned the first value before the first loop; the second value before the second loop; the third value before the third loop, and so forth. The statement block uses the value of <i>ControlVariable</i> to direct macro execution.</p> <p>ENDFOR closes a FOREACH statement.</p> |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                        |                                                                                             |
|------------------------|---------------------------------------------------------------------------------------------|
| <b>Parameters</b>      |                                                                                             |
| <i>ControlVariable</i> | Variable: Control variable.                                                                 |
| <i>ValueList</i>       | Any: Variables, constants, or expressions, enclosed in braces and separated by a semicolon. |

---

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre>APPLICATION(A1; "WordPerfect"; Default; "US") // Beep vCount times - BREAK when vCount equals 4 // Remember: functions do not execute unless called // Demonstrate FOREACH command  FUNCTION BeepBeep(vCount)     REPEAT         BEEP         WAIT(3)         vCount := vCount - 1     UNTIL(vCount = 0)     RETURN ENDFUNC  FOREACH(vCount; {1; 2; 3; 4; 5})     IF(vCount = 4)         BREAK     ENDIF     BeepBeep(vCount)     WAIT(5) ENDFOR  MESSAGEBOX(x; "BREAK"; "Variable vCount equals 4"; IconInformation!) QUIT</pre> |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**See Also**

♦ FOR ♦ FORNEXT ♦ REPEAT ♦ WHILE

---

## FORNEXT-ENDFOR

---

**Purpose**

A loop statement that executes a specified number of times. See *Loop Statements* in *Macro Control Statements*.

---

**Syntax**

The general form of a FORNEXT statement is,

```
FORNEXT (ControlVariable; InitialValue; FinalValue; IncrementValue)
...statement block...
ENDFOR
```

The following statement initializes *vTest* to one, repeats the statement block while *vTest* is less than or equal to five, and increments *vTest* by two after each loop.

```
FORNEXT(vTest; 1; 5; 2)
...statement block...
ENDFOR
```

---

**Description**

The loop executes if *ControlVariable* is less than or equal to *FinalValue*, and executes until *ControlVariable* is greater than *FinalValue*. If you use a negative value in *IncrementValue*, the loop executes until *ControlVariable* is less than *FinalValue*.

ENDFOR closes a FORNEXT statement.

---

**Parameters**

*ControlVariable*

Variable: Control variable.

*InitialValue*

Any: The value assigned to *ControlVariable* at the start of a loop.

*FinalValue*

Any: A variable, constant, or expression. The loop executes while *FinalValue* is greater than *ControlVariable*, or less than if *IncrementValue* is a negative value.

*IncrementValue*

Any (optional): The value to increment *ControlVariable* after each loop. Default is 1.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Execute Windows calculator
// Locate and activate 3 times
// Demonstrate FORNEXT command

ASSIGN(hWP; APPLOCATE(""))
APPEXECUTE("c:\windows\calc.exe")
ASSIGN(hCalc; APPLOCATE("Calculator"))
FORNEXT(vCtrl; 1; 2; 1)
 WAIT(10)
 APPACTIVATE(hWP)
 WAIT(10)
 APPACTIVATE(hCalc)
ENDFOR
```

---

**See Also**

♦ FOR ♦ FOREACH ♦ REPEAT ♦ WHILE

---

# FRACTION

---

**Purpose**

Returns the fractional portion of a real number.

---

**Syntax**ReturnValue := FRACTION (*Value*)

---

**Returns**

A fraction.

```
vFraction := FRACTION(1.5) // vFraction = 0.5
```

Result: vFraction equals 0.5

```
vResult := FRACTION(1.77 * 2) // vResult = 0.54
```

Result: vResult equals 0.54

---

**Parameters***Value*

Numeric Expression: A real number.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Calculate fractions and integers
// Demonstrate FRACTION and INTEGER commands

ONCANCEL(QuitMacro)
ASSIGN(HdReturn; NTOC(0F90Ah)) // hard return code

REPEAT
 GETNUMBER(vNnbr; "Enter a real number: "; "FRACTION EXAMPLE")

 ASSIGN(vInteger; INTEGER(vNnbr))
 ASSIGN(vFraction; FRACTION(vNnbr))

 vMsg1 := "The integer portion of " + vNnbr + " is " + vInteger
 vMsg2 := "The fractional portion of " + vNnbr + " is " + vFraction

 MESSAGEBOX(x; "INTEGER - FRACTION"; vMsg1 + HdReturn + HdReturn + vMsg2 +
 HdReturn; IconInformation! | RetryCancel!)
UNTIL(x = 2)

LABEL(QuitMacro)
QUIT
```

---

**See Also**

♦ INTEGER

---

# FUNCTION

---

## Purpose

Identifies a macro subroutine that can receive one or more values from a calling statement (see *Calling Statements* in *Macro Control Statements*). FUNCTION also returns a value to the calling statement (caller).

## Syntax

The general form of a FUNCTION statement is,

```
FUNCTION Name (Parameter; Parameter; ... Parameter)
...statement block...
ENDFUNC
```

## Description

Functions contain one or more statements that execute when the function is called. Unlike LABEL statements, functions do not execute unless called. A calling statement consists of the function's name, and can have one or more parameters that pass values to the function. RETURN or ENDFUNC direct macro execution to the statement that follows the function's caller. RETURN also returns the result of a function operation, or 0 if no result is specified. For example,

```
y := 5
x := Add(y) // calling statement

FUNCTION Add(z)
 z := z + 5
 RETURN(z)
ENDFUNC
```

assigns the value 10 to variable x. Add is the name of the function, and y contains a value to pass. Variable z has no value until the function is called, when the value of z is 5 (the value of variable y). The first function statement assigns 10 to variable z. The next statement returns the value of z to the calling statement, which is then assigned to variable x. If the function did not contain a RETURN statement, or if RETURN did not return a value, x would be equal to 0.

### Address Mode

In the above example, the value of y does not change. The function returns the value of z in variable x. To change the value of a variable passed to a function, precede the calling statement parameter and its corresponding function parameter with an ampersand (&). For example,

```
y := 5
x := Add(&y)

FUNCTION Add(&z)
 ASSIGN(z; z + 5)
 RETURN(z)
ENDFUNC
```

assigns the value 10 to both x and y. The ampersand before variable x means the variable's address (location in memory) is passed to the function, not the variable's value. Changes made at the address of x are made to the contents of x.

## Arrays

You pass arrays to functions the same way you pass variables. If you are passing the entire array, every array element must be assigned a value. If not, a run-time error identifies a reference to the undefined element. For example,

```
DECLARE w[10]
FORNEXT(x; 1; 9; 1) // assign only 9 elements
 w[x] = x
ENDFOR

FUNCTION Test(z[])
 FORNEXT(x; 1; 10; 1)
 x[z] = x * 10
 ENDFOR
 RETURN(z[]) // the value of 10 elements returned
ENDFUNC

y[] = Test(w[]) // 10 elements returned in array y[]

Type(y[10]) // y[10] equals 100
Type(w[10]) // Run-time error: Undefined variable "W[10]"
```

In the previous example, if you precede the calling statement parameter and the corresponding function parameter with an ampersand (&), 100 is returned in w[10]. No run-time error occurs. You are passing the address of array w[], not its value. The value is assigned inside the function (see *Address Mode* above). The two statements look like this:

```
FUNCTION Test(&z[])

and

y[] = Test(&w[])
```

Pass the value of an array element, the same way you pass a variable. For example,

```
DECLARE w[10]
FORNEXT(x; 1; 10; 1)
 w[x] = x
ENDFOR

FUNCTION Test(z)
 z := z * 10
 RETURN(z)
ENDFUNC

y = Test(w[1])

Type(y + " - ")
Type(w[1])
```

The value of y equals 10. The value of w[1] equals 1. If you precede the calling statement parameter and the corresponding function parameter with an ampersand

(&z), the value of y equals 10 and the value of w[1] equals 10. Passing the address of w[1] causes its value to change inside the function. The two statements look like this:

```
FUNCTION Test(&z)
```

*and*

```
y = Test(&w[1])
```

---

## Parameters

### *Name*

Label: The name of a function. It begins with a letter and consists of one or more letters or numbers.

### *Parameter*

Variable: Receives a value from a calling statement (see *Calling Statements in Macro Control Statements*). If an ampersand precedes the calling statement variable, an ampersand must precede the corresponding function variable (see *Address Mode* above). Multiple variables are separated by a semicolon.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Remember: functions must be called to execute
// Demonstrate FUNCTION command
```

```
HdReturn := NTOC(0F90Ah)
```

```
x := 4
```

```
vBeep := 1
```

```
FUNCTION Check(vBeep; HdReturn)
```

```
 MESSAGEBOX(vStatus; "FUNCTION EXAMPLE"; "Beeps: " + vBeep + HdReturn + HdReturn +
 "Choose Retry to beep again." + HdReturn; IconInformation! | RetryCancel!)
```

```
 RETURN(vStatus)
```

```
ENDFUNC
```

```
WHILE(x = 4)
```

```
 BEEP
```

```
 x := Check(vBeep; HdReturn)
```

```
 vBeep := vBeep + 1
```

```
ENDWHILE
```

```
MESSAGEBOX(x; "RETURN"; "The value of variable vStatus (" + vStatus + ") is returned to variable x,
which ends the loop."; IconExclamation!)
```

---

## See Also

♦ CALL ♦ LABEL ♦ PROCEDURE ♦ RETURN ♦ USE



---

# GETNUMBER

---

**Purpose** Displays a dialog box that contains an edit control to enter an integer or real number.

---

**Syntax** GETNUMBER (*MacroVar*; *Prompt*; *Title*)

---

## Parameters

*MacroVar* Variable: The number entered in the edit box is returned in this variable.

*Prompt* Character Expression (optional): Text displayed above the edit control. Use NTOC(0F90Ah) to insert a hard return code in the prompt string. For example,

```
GETNUMBER(vAns; "First line" + NTOC(0F90Ah) + "Second line"; "Title")
```

creates two lines above the edit box. The next example creates three lines, including one blank line:

```
ASSIGN(HdReturn; NTOC(0F90Ah))
GETNUMBER(vAns; "First line" + HdReturn + HdReturn + "Second line"; "Title")
```

*Title* Character Expression (optional): Text displayed in the title bar.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input
// Remember: procedures do not execute unless called
// Demonstrate GETNUMBER command

PROCEDURE ErrorMsg(x)
 BEEP BEEP BEEP
 IF(x < 1)
 vMessage := x + " is less than 1"
 ELSE
 vMessage := x + " is greater than 5"
 ENDIF
 MESSAGEBOX(vStatus; "OUT-OF-RANGE ERROR"; vMessage + " - please try again";
 IconExclamation!)
ENDPROC

REPEAT
 GETNUMBER(vNum; "Enter a number from 1 to 5"; "GETNUMBER EXAMPLE")
 IF((vNum < 1) OR (vNum > 5))
 CALL ErrorMsg(vNum)
 ELSE
 MESSAGEBOX(vStatus; "YES!"; "You entered " + vNum; IconInformation!)
 ENDIF
UNTIL((vNum > 0) AND (vNum < 6))
```

---

**See Also** ♦ GETSTRING ♦ GETUNITS

---

# GETSTRING

---

**Purpose** Displays a dialog box that contains an edit control to enter a character string.

---

**Syntax** GETSTRING (*MacroVar*; *Prompt*; *Title*; *Length*)

---

## Parameters

*MacroVar* Variable: The character string entered in the edit box is returned in this variable.

*Prompt* Character Expression (optional): The text displayed above the edit control. Use NTOC(0F90Ah) to insert a hard return code in the prompt string. For example,

```
GETSTRING(vAns; "First line" + NTOC(0F90Ah) + "Second line"; "Title")
```

creates two lines above the edit box. The next example creates three lines, including one blank line:

```
ASSIGN(HdReturn; NTOC(0F90Ah))
GETSTRING(vAns; "First line" + HdReturn + HdReturn + "Second line"; "Title")
```

*Title* Character Expression (optional): The text displayed in the title bar.

*Length* Numeric Expression (optional): The maximum number of characters the edit control accepts. Default is unlimited.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract string of numbers and convert to numeric equivalent
// Demonstrate GETSTRING command

vNumbers := ""
vPrompt := "Enter text that includes both letters and numbers:"
LABEL(Start)
GETSTRING(vStr; vPrompt; "EXTRACT NUMBERS")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vLen := STRLEN(vStr)
vPos := 1
REPEAT
 vTest := SUBSTR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest
 ENDIF
 vPos := vPos + 1
UNTIL(vPos = vLen + 1)
IF(STRLEN(vNumbers) > 0)
 vConvert := STRNUM(vNumbers)
 vConvert := vConvert * 5
 MESSAGEBOX(x; "STRNUM COMMAND"; vNumbers + " * 5 equals " + vConvert;
 IconInformation!)
```

```
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "You didn't enter any numbers"; IconExclamation!)
ENDIF
```

---

**See Also** ♦ GETNUMBER ♦ GETUNITS

---

## GETUNITS

---

**Purpose** Displays a dialog box that contains an edit control to enter a number and unit of measure.

---

**Syntax** GETUNITS (*MacroVar*; *Prompt*; *Title*)

---

**Description** The units of measure are " (inches); i (inches); c (centimeters); m (millimeters); p (points—72 per inch); and w (WordPerfect units—1200 per inch). If a unit of measure is not specified, the default is WordPerfect units. To change the default use DEFAULTUNITS.

---

### Parameters

*MacroVar* Variable: The number and unit of measure entered in the edit box is returned in this variable. If no unit of measure is specified, the default is returned.

*Prompt* Character Expression (optional): The text displayed above the edit control. Use NTOC(0F90Ah) to insert a hard return code in the prompt string. For example,

```
GETUNITS(vAns; "First line" + NTOC(0F90Ah) + "Second line"; "Title")
```

creates two lines above the edit box. The next example creates three lines, including one blank line:

```
ASSIGN(HdReturn; NTOC(0F90Ah))
GETUNITS(vAns; "First line" + HdReturn + HdReturn + "Second line"; "Title")
```

*Title* Character Expression (optional): The text displayed in the title bar.

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Specify a unit of measure or accept WP units default
// Demonstrate GETUNITS command

MENU(vChoice; Digit;;; {"Advance down"; "Advance up"; "Advance left"; "Advance right"})
SWITCH(vChoice)
 CASEOF 1: vPrompt := "Advance down:"
 CASEOF 2: vPrompt := "Advance up:"
 CASEOF 3: vPrompt := "Advance left:"
 CASEOF 4: vPrompt := "Advance right:"
ENDSWITCH
DEFAULTUNITS(Inches!)
```

```
GETUNITS(vUnit; vPrompt; "GETUNITS EXAMPLE")
SWITCH(vChoice)
 CASEOF 1: ADVANCE(AdvanceDown!; vUnit)
 CASEOF 2: ADVANCE(AdvanceUp!; vUnit)
 CASEOF 3: ADVANCE(AdvanceLeft!; vUnit)
 CASEOF 4: ADVANCE(AdvanceRight!; vUnit)
ENDSWITCH
```

---

**See Also**

♦ DEFAULTUNITS ♦ GETSTRING ♦ GETNUMBER ♦ UNITSTR

---

## GLOBAL

---

**Purpose**

Declares global variables and arrays, and assigns them to the global variable table.

---

**Syntax**

GLOBAL (*VariableName*; *VariableName*; ... *VariableName*)

---

**Description**

Global variables can be used in RUN or CHAIN macros.

---

**Parameters**

*VariableName*

Variable: One or more user-defined variables separated by a semicolon. Variables must begin with a letter and can include any other combination of letters or numbers. Parentheses are optional. For example,

```
GLOBAL VariableName; VariableName
GLOBAL(VariableName; VariableName)
```

are both valid statements. The next example declares global arrays with 10 elements and 20 elements:

```
GLOBAL FirstArray[10]; NextArray[20]
```

---

**Example**

```
// There are four macros in this example
// Compile separately

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro1.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro2.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro3.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)
```

```

APPLICATION(A1: "WordPerfect"; Default; "US")
// Calling Macro
// Demonstrate CHAIN command and GLOBAL variables

GLOBAL vTitle; vPrompt
MENU(vChoice; Letter;;; {"Macro 1"; "Macro 2"; "Macro 3"})
ASSIGN(vPrompt; "Computer beeps before chain macro called")
ASSIGN(vMacrosPath; ?PathMacros)
SWITCH(vChoice)
 CASEOF 1:
 ASSIGN(vTitle; "MACRO ONE")
 CHAIN(vMacrosPath + "macro1.wcm")
 CASEOF 2:
 ASSIGN(vTitle; "MACRO TWO")
 CHAIN(vMacrosPath + "macro2.wcm")
 CASEOF 3:
 ASSIGN(vTitle; "MACRO THREE")
 CHAIN(vMacrosPath + "macro3.wcm")
ENDSWITCH
// beep sounds before chain macro called
BEEP WAIT(5)

```

---

### See Also

♦ CHAIN ♦ DECLARE ♦ DISCARD ♦ EXISTS ♦ LOCAL ♦ PERSIST  
♦ PERSISTALL ♦ RUN

---

## GO

|                    |                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Jumps to a LABEL statement and does not return.                                                     |
| <b>Syntax</b>      | Go ( <i>Label</i> )                                                                                 |
| <b>Description</b> | Go is generally used to exit multiple layers of nested statements, or to create a loop.             |
| <b>Parameters</b>  |                                                                                                     |
| <i>Label</i>       | Label: The name of a label. It begins with a letter and consists of one or more letters or numbers. |

|                |                                                                                                                                                                                                                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> APPLICATION(A1: "WordPerfect"; Default; "US") // Extract string of numbers and convert to measurement equivalent // Demonstrate GO command  Retry := 4 LABEL(Start)   vNumbers := ""   GETSTRING(vStr; "Enter text that includes both letters and numbers."; "EXTRACT NUMBERS")   IF(vStr = "")     BEEP </pre> |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
 ENDIF
 vLen := STRLEN(vStr)
 vPos := 1
 REPEAT
 vTest := SUBSTR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest
 ENDIF
 vPos := vPos + 1
 UNTIL(vPos = vLen + 1)
 IF(STRLEN(vNumbers) > 0)
 CALL(DefaultUnit)
 vConvert := STRUNIT(vNumbers)
 vConvert := vConvert * 5
 MESSAGEBOX(x; "STRUNIT COMMAND"; vNumbers + " * 5 equals " + vConvert;
 IconInformation! | RetryCancel!)
 IF(x = Retry)
 GO(Start)
 ENDIF
 ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "You didn't enter any numbers"; IconExclamation!)
 ENDIF
 QUIT

LABEL(DefaultUnit)
 MENU(vChoice; Digit;;; {"Inches"; "Centimeters"; "Millimeters"; "Points"; "WP Units"})
 SWITCH(vChoice)
 CASEOF 1: DEFAULTUNITS(Inches!)
 CASEOF 2: DEFAULTUNITS(Centimeters!)
 CASEOF 3: DEFAULTUNITS(Millimeters!)
 CASEOF 4: DEFAULTUNITS(Points!)
 CASEOF 5: DEFAULTUNITS(WPUnits!)
 ENDSWITCH
 RETURN

```

---

## See Also

♦ CALL ♦ LABEL

---

## IF-ELSE-ENDIF

### Purpose

A conditional statement that determines whether a statement (or statement block) is executed. See *Conditional Statements in Macro Control Statements*.

---

**Syntax**                    The general form of an IF statement is,

```
IF (Test)
...statement block...
ELSE
...statement block...
ENDIF
```

---

**Description**            If *Test* is true, the statements between IF and ELSE are executed. If false, the statements between ELSE and ENDIF are executed.

ELSE is optional. If *Test* is false and ELSE is not used, the macro skips the statements between IF and ENDIF, and executes the first statement after ENDIF.

ENDIF closes an IF statement.

---

**Parameters**

*Test*                        Relational Expression: Evaluates to true or false.

---

**Example**

```
APPLICATION(A1; "WordPerfect": Default; "US")
// Test user input
// Remember: procedures do not execute unless called
// Demonstrate IF command

PROCEDURE ErrorMsg(x)
 BEEP BEEP BEEP
 IF(x < 1)
 vMessage := x + " is less than 1"
 ELSE
 vMessage := x + " is greater than 5"
 ENDIF
 MESSAGEBOX(vStatus; "OUT-OF-RANGE ERROR"; vMessage + " - please try again";
 IconExclamation!)
ENDPROC

REPEAT
 GETNUMBER(vNum; "Enter a number from 1 to 5"; "GETNUMBER EXAMPLE")
 IF((vNum < 1) OR (vNum > 5))
 CALL ErrorMsg(vNum)
 ELSE
 MESSAGEBOX(vStatus; "YES!"; "You entered " + vNum; IconInformation!)
 ENDIF
UNTIL((vNum > 0) AND (vNum < 6))
```

---

**See Also**                    ♦ CASE ♦ CASE CALL ♦ SWITCH

---

---

# IFPLATFORM-ENDIFPLATFORM

---

**Purpose** A conditional statement that specifies a platform or platforms for which subsequent statements are compiled.

---

**Syntax** The general form of an IFPLATFORM statement is,  
IFPLATFORM (*PlatformID*; *PlatformID*; ... *PlatformID*)  
...*statement block*...  
ENDIFPLATFORM

---

**Description** If the current platform matches one of the specified platforms, the statements between IFPLATFORM and ENDIFPLATFORM are compiled and executed.  
ENDIFPLATFORM closes an IFPLATFORM statement.

---

## Parameters

*PlatformID* Enumeration: One or more platform identifiers. Separate multiple platforms with a semicolon. The identifiers for DOS and Windows are DOS and WIN. See platform documentation for other identifiers.

---

## Example

```
// Create a DOS or Windows dialog box
// Demonstrate IFPLATFORM command

// compile on DOS platform
IFPLATFORM(DOS)
 DLGCREATE(x; "TITLE"; DLGNoCancel!;;30;14)
 DLGCONTROL(CtrlRadioButton!;a;"Letter";StylInitial!;4;4;;2)
 DLGCONTROL(CtrlRadioButton!;b;"Itinerary";;4;6;;2)
 DLGCONTROL(CtrlRadioButton!;c;"Memo";;4;8;;2)
 DLGCONTROL(CtrlRadioButton!;d;"Fax";;4;10;;2)
DLGEND

SWITCH(1)
 CASEOF a: BEEP
 CASEOF b: BEEP
 CASEOF c: BEEP
 CASEOF d: BEEP
ENDSWITCH
ENDIFPLATFORM

// compile on Windows platform
IFPLATFORM(WIN)
 APPLICATION(A1; "WordPerfect"; Default; "US")
 DIALOGDEFINE(1000; 50; 50; 100; 125; Style:OK!; Caption:"TITLE")
 DIALOGADDRADIOBUTTON(1000; 1; 10; 10; 75; 15; ButtonText:"Letter"; a)
 DIALOGADDRADIOBUTTON(1000; 2; 10; 25; 75; 15; ButtonText:"Itinerary"; b)
 DIALOGADDRADIOBUTTON(1000; 3; 10; 40; 75; 15; ButtonText:"Memo"; c)
 DIALOGADDRADIOBUTTON(1000; 4; 10; 55; 75; 15; ButtonText:"Fax"; d)
 DIALOGDISPLAY(1000; 1)
```



```

SWITCH(1)
 CASEOF a: vMsg := "Letter"
 CASEOF b: vMsg := "Itinerary"
 CASEOF c: vMsg := "Memo"
 CASEOF d: vMsg := "Fax"
ENDSWITCH
MESSAGEBOX(x; "Selection"; vMsg; IconInformation!)
DIALOGDESTROY(1000)
ENDIFPLATFORM

```

---

## INDIRECT

---

**Purpose** Creates variable and label names out of a combination of character strings and/or numbers.

---

**Syntax** ReturnValue := INDIRECT (*VariableName*)

---

**Returns** The contents of *VariableName*.

---

**Description** You can use INDIRECT wherever you use a variable. The actual variable must be declared and initialized before INDIRECT. You can use INDIRECT to call a LABEL, but not to create one.

```

State1 := "Utah"
x := INDIRECT("State" + 1)

```

Result: x equals Utah

### Variable example in shorthand notation

```

State1 := "Utah"
State2 := "Idaho"
State3 := "Arizona"

FORNEXT(Nmbr; 1; 3; 1)
 Type("State of " + INDIRECT("State" + Nmbr))
 HardReturn
ENDFOR

```

Explanation: Types three lines, State of Utah, State of Idaho, and State of Arizona.

### LABEL example

```

FORNEXT(x; 1; 3; 1)
 CALL(INDIRECT("Lab" + x))
ENDFOR
QUIT

```

```
LABEL(Lab1)
...statement block...
RETURN
```

```
LABEL(Lab2)
...statement block...
RETURN
```

```
LABEL(Lab3)
...statement block...
RETURN
```

Explanation: Call three Labels with a single CALL statement in a FORNEXT loop.

---

## Parameters

*VariableName*

Character Expression: Character strings and/or numbers are concatenated to form variable and Label names. You cannot use INDIRECT to create a subroutine name.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Declare and test variable types
// Demonstrate INDIRECT, BREAK, EXIST, LOCAL, GLOBAL, and PERSIST commands

DISCARD var5; var1; var7
LOCAL var2; var9; var4
GLOBAL var3; var10; var6
PERSIST var5; var1; var7
ASSIGN(var8; "")

FOR(x; 1; x < 11; x + 1)
 ASSIGN(INDIRECT("var" + x); x)
 MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion! | YesNo!)
 IF(vStatus = 7)
 BREAK
 ENDIF
ENDFOR

FOR(z; 1; z < (x + 1); z + 1)
 SWITCH(EXISTS(INDIRECT("var" + z)))
 CASEOF 1: Msg("Local"; z)
 CASEOF 2: Msg("Global"; z)
 CASEOF 3: Msg("Persist"; z)
 ENDSWITCH
ENDFOR

FUNCTION Msg(w; y)
 MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w)
ENDFUNC
```

---

**See Also**

♦ ASSIGN

---

# INTEGER

---

**Purpose** Returns the integer portion of a real number.

---

**Syntax** Return*Value* := INTEGER (*Value*)

---

**Description** If *Value* does not contain an integer, 0 is returned. A negative real number is rounded up to the next integer value, and a positive real number is rounded down.

---

**Returns** An integer.

```
vInteger := INTEGER(1.5)
Result: vInteger equals 1
vResult := INTEGER(1.77 * 2)
Result: vResult equals 3
vZero := INTEGER(.7)
Result: vZero equals 0
vNegative := INTEGER(-1.77)
Result: vNegative equals -2
```

---

**Parameters***Value* Numeric Expression: A real number.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Calculate fractions and integers
// Demonstrate FRACTION and INTEGER commands

ONCANCEL(QuitMacro)
ASSIGN(HdReturn; NTOC(0F90Ah)) // hard return code

REPEAT
 GETNUMBER(vNbr; "Enter a real number: "; "FRACTION EXAMPLE")

 ASSIGN(vInteger; INTEGER(vNbr))
 ASSIGN(vFraction; FRACTION(vNbr))

 vMsg1 := "The integer portion of " + vNbr + " is " + vInteger
 vMsg2 := "The fractional portion of " + vNbr + " is " + vFraction

 MESSAGEBOX(x; "INTEGER - FRACTION"; vMsg1 + HdReturn + HdReturn + vMsg2 +
 HdReturn; IconInformation! | RetryCancel!)
UNTIL(x = 2)

LABEL(QuitMacro)
QUIT
```

---

**See Also**

♦ FRACTION

---

**LABEL**

---

**Purpose**

Identifies a macro subroutine, which generally includes a statement block followed by RETURN or QUIT. See *Subroutines* in *Macro Control Statements*.

---

**Syntax**LABEL (*Label*)

---

**Description**

LABEL is used by CALL, CASE, CASE CALL, GO, ONCANCEL, ONCANCEL CALL, ONERROR, ONERROR CALL, ONNOTFOUND, ONNOTFOUND CALL, ONDDEADVISE CALL, DDEEXECUTEEXT.

---

**Parameters***Label*

Label: The name of a label. It begins with a letter and consists of one or more letters or numbers.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate LABEL command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
MENU(vPick; Digit!;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
SWITCH(vPick)
CASEOF 1: ASSERT(CancelCondition!)
CASEOF 2: ASSERT(ErrorCondition!)
CASEOF 3: ASSERT(NotFoundCondition!)
CASEOF 4: QUIT
DEFAULT: GO(Condition)
ENDSWITCH

LABEL(Condition)
SWITCH(ErrorNumber)
CASEOF 1: vMsg := "1 (Cancel condition)"
CASEOF 2: vMsg := "2 (Error condition)"
CASEOF 7: vMsg := "3 (Not Found condition)"
DEFAULT: vMsg := "?"
ENDSWITCH
MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
GO(Start)
```

---

**See Also**

♦ CALL ♦ CASE ♦ CASE CALL ♦ DDEEXECUTEEXT ♦ GO ♦ ONCANCEL  
♦ ONCANCEL ♦ CALL ♦ ONDDEADVISE CALL ♦ ONERROR ♦ ONERROR  
CALL ♦ ONNOTFOUND ♦ ONNOTFOUND CALL

---

# LOCAL

---

**Purpose** Declares local variables and arrays, and assigns them to the local variable table.

---

**Syntax** LOCAL (*VariableName*; *VariableName*; ... *VariableName*)

---

**Description** Variables are LOCAL by default. The Local variable table is removed from memory when the macro ends.

---

## Parameters

*VariableName* Variable: One or more user-defined variables separated by a semicolon. Variables must begin with a letter and can include any other combination of letters or numbers. Separate multiple variables with a semicolon. Parentheses are optional. For example,

```
LOCAL VariableName; VariableName
LOCAL(VariableName; VariableName)
```

are both valid statements. The next example declares local arrays with 10 elements and 20 elements:

```
LOCAL FirstArray[10]; NextArray[20]
```

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Declare and test variable types
// Demonstrate BREAK, EXIST, LOCAL, GLOBAL, and PERSIST commands

DISCARD var5; var1; var7
LOCAL var2; var9; var4
GLOBAL var3; var10; var6
PERSIST var5; var1; var7
ASSIGN(var8; "")

FOR(x; 1; x < 11; x + 1)
 ASSIGN(INDIRECT("var" + x); x)
 MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion! | YesNo!)
 IF(vStatus = 7)
 BREAK
 ENDIF
ENDFOR

FOR(z; 1; z < (x + 1); z + 1)
 SWITCH(EXISTS(INDIRECT("var" + z)))
 CASEOF 1: Msg("Local"; z)
 CASEOF 2: Msg("Global"; z)
```

```

 CASEOF 3: Msg("Persist"; z)
 ENDSWITCH
 ENDFOR

 FUNCTION Msg(w; y)
 MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w)
 ENDFUNC

```

---

**See Also**

♦ DECLARE ♦ DISCARD ♦ EXISTS ♦ GLOBAL ♦ PERSIST ♦ PERSISTALL

---

## MENU

---

**Purpose**

Displays a menu of user-defined items.

---

**Syntax**

MENU (*MacroVar*; *MenuType*; *HorizPosition*; *VertiPosition*; {*MenuText*; *MenuText*; ...})

---

**Description**

MENU is used with statements such as CASE, CASE CALL, and IF to execute a statement that corresponds to a selected menu item. To dismiss a menu, press Alt or Esc.

---

**Parameters**
*MacroVar*

Variable: A number is returned in this variable, whether numbers or letters are used to reference menu items (see *MenuType* parameter). For numbers, the range is 1 to 9. For letters, the range is 1 to 26. A returns 1, B returns 2, and so forth. 0 is returned if Alt or Esc is pressed.

*MenuType*

Enumeration: Determines whether a letter or number is displayed to the left of a menu item. The enumerations are,

```

 Digit! Numbers (1-9)
 Letter! Alphabetic characters (A-Z)

```

*HorizPosition*

Numeric Expression (optional): The number of pixels from the left side of the main window to the left side of the menu. To center the menu horizontally, leave this parameter blank.

*VertiPosition*

Numeric Expression (optional): The number of pixels from the top of the main window to the top of the menu. To center the menu vertically, leave this parameter blank.

*MenuText*

Character Expression: The menu item text. You may list up to nine menu items referenced by numbers, or up to twenty-six referenced by letters. Enclose menu items in braces ({}), separated by a semicolon.

```
MENU(vSelection; Letters; 50; 50; {"Option 1"; "Option 2"; "Option 3"})
```

To select an item: click it, select it with Tab and press Enter, or press the corresponding letter or number.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate MENU command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
MENU(vPick; Digit;;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
SWITCH(vPick)
CASEOF 1: ASSERT(CancelCondition!)
CASEOF 2: ASSERT(ErrorCondition!)
CASEOF 3: ASSERT(NotFoundCondition!)
CASEOF 4: QUIT
DEFAULT: GO(Condition)
ENDSWITCH

LABEL(Condition)
SWITCH(ErrorNumber)
CASEOF 1: vMsg := "1 (Cancel condition)"
CASEOF 2: vMsg := "2 (Error condition)"
CASEOF 7: vMsg := "3 (Not Found condition)"
DEFAULT: vMsg := "?"
ENDSWITCH
MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
GO(Start)
```

---

**See Also**

♦ CASE ♦ CASE CALL ♦ IF

---

## MENULIST

---

**Purpose**

Displays a menu of user-defined items.

---

**Syntax**

*MENULIST (VariableName; {MenuChoice; MenuChoice; ...}; Title, HorizPosition; VertiPosition)*

---

**Description**

This command is included for DOS compatibility. The Windows equivalent is MENU.

MENULIST is used with statements such as CASE, CASE CALL, and IF to execute a statement that corresponds to a selected menu item. To dismiss a menu, press Alt or Esc.

---

## Parameters

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>VariableName</i> | Variable: A number is returned in this variable, whether numbers or letters are used to reference menu items (see <i>MenuChoice</i> parameter). For numbers, the range is 1 to 9. For letters, the range is 1 to 26. A returns 1, B returns 2, and so forth. 0 is returned if Alt or Esc is pressed.                                                                                                                                     |
| <i>MenuChoice</i>   | Character Expression: The menu item text, enclosed in braces ({}), and separated by a semicolon.<br><br>MENULIST(vSelection;;; {"Option 1"; "Option 2"; "Option 3"})<br><br>To select an item: click it, select it with Tab and press Enter, or press the corresponding letter or number. Nine items or less are displayed with a number to the left of the menu item. More than nine are displayed with a letter. The range is 1 to 26. |
| <i>Title</i>        | Character Expression: The text displayed in the title bar.                                                                                                                                                                                                                                                                                                                                                                               |

---

**Example** See MENU

---

**See Also** ♦ CASE ♦ CASE CALL ♦ IF ♦ MENU

---

# MESSAGEBOX

---

**Purpose** Displays a message box.

---

**Syntax** MESSAGEBOX (*MacroVar*; *Title*; *Message*; *Style*; *ParameterData*)

---

**Description** Provides a limited set of buttons and icons, and an option to include up to ten different parameter strings in the message.

---

## Parameters

|                 |                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>MacroVar</i> | Variable: The control value of the button that dismisses the message box is returned in this variable. The return button values are,<br><br><ol style="list-style-type: none"><li>1 OK</li><li>2 Cancel</li><li>3 Abort</li><li>4 Retry</li><li>5 Ignore</li><li>6 Yes</li><li>7 No</li></ol> |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

If there is not enough memory to create the message box, 0 is returned.



## Title

Character Expression: The title bar text.

## Message

Character Expression: The message box text. Icons appear to the left of the message. Control buttons are centered below the message. If `HasParameters!` style is used, a caret (^) followed by a number inserts the corresponding *ParameterData* message in its place (see *ParameterData* parameter below). Use two carets (^) to insert a caret as part of the message string. Use `NTOC(0F90Ah)` to insert a hard return code in the message string. For example,

```
MESSAGEBOX(vAns: "Title"; "First line" + NTOC(0F90Ah) + "Second line"; IconInformation!)
```

creates two lines above the control button. The next example creates three lines, including one blank line:

```
ASSIGN(HdReturn; NTOC(0F90Ah))
MESSAGEBOX(vAns: "Title"; "First line" + HdReturn + HdReturn + "Second line"; IconInformation!)
```

## Style

Enumeration: Message box styles. Type | between enumerations to combine styles. Select only one style from each group. Button enumerations are,

|                                |                                        |
|--------------------------------|----------------------------------------|
| <code>AbortRetryIgnore!</code> | Abort, Retry, and Ignore               |
| <code>OK!</code>               | OK (default if no button is specified) |
| <code>OKCancel!</code>         | OK and Cancel                          |
| <code>RetryCancel!</code>      | Retry and Cancel                       |
| <code>YesNo!</code>            | Yes and No                             |
| <code>YesNoCancel!</code>      | Yes, No, and Cancel                    |

Icon enumerations are:

|                               |                                           |
|-------------------------------|-------------------------------------------|
| <code>IconNone!</code>        | No icon (default if no icon is specified) |
| <code>IconExclamation!</code> | Exclamation point in a yellow circle      |
| <code>IconInformation!</code> | Lowercase "i" in a blue circle            |
| <code>IconAsterisk!</code>    | Lowercase "i" in a blue circle            |
| <code>IconQuestion!</code>    | Question mark in a green circle           |
| <code>IconStop!</code>        | Stop sign in a red circle                 |
| <code>IconHand!</code>        | Stop sign in a red circle                 |

Modality enumerations are:

|                                |                                                                                                                                                                                                                                                               |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ApplicationModal!</code> | You can switch to another application, but you must dismiss the message box (click a control button) before the macro resumes. <code>ApplicationModal!</code> is the default if either <code>SystemModal!</code> or <code>TaskModal!</code> is not specified. |
| <code>SystemModal!</code>      | You must dismiss the message box before the macro resumes, and before you can switch to another application.                                                                                                                                                  |
| <code>TaskModal!</code>        | You can switch to another application, but you must dismiss the message box (click a control button) before the macro resumes.                                                                                                                                |

Miscellaneous enumerations are:

|                    |                                         |
|--------------------|-----------------------------------------|
| <code>Beep!</code> | Beep when the message box is displayed. |
|--------------------|-----------------------------------------|

HasParameters!      Notifies MessageBox box that a caret (^) followed by a number in the *Message* parameter is to be replaced by the corresponding character string in the *ParameterData* parameter. See ASSERT.WCM example (lines 44-50) in *Making Decisions with MENU*.

### ParameterData

Character Expression: The text for the *Message* parameter (numbering begins with 0). Up to ten parameters separated by semicolons are allowed. For example,

```
v0 := "Yes to continue, or"
v1 := "No to quit."
MessageBox(x; "Title"; "Select ^0 ^1"; YesNo! | IconStop! | HasParameters!; {v0; v1})
```

fills the *Message* parameter with "Select Yes to continue, or No to quit.". If HasParameters! is not used, *Message* contains "Select ^0 ^1".

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Create a loop with GO command
// Demonstrate CASE, MESSAGEBOX, and system variables

LABEL(Start)
 MENU(vChoice; Digit;;; {"IconExclamation!"; "IconInformation!"; "IconQuestion"; "IconStop!";
 "Quit"})
 CASE(vChoice; {1; Exclaim; 2; Info; 3; Question; 4; Stop; 5; QuitMacro}; QuitMacro)

LABEL(Exclaim)
 MESSAGEBOX(x; "EXCLAMATION!"; "Example of an exclamation icon"; IconExclamation!)
 GO(LoopToStart)

LABEL(Info)
 ASSIGN(vDay; ?DateWeekday)
 MESSAGEBOX(x; "INFORMATION!"; "Example of an information icon - The current day of the
 week is " + vDay + "."; IconInformation!)
 GO(LoopToStart)

LABEL(Question)
 Yes := 6
 MESSAGEBOX(vAns; "QUESTION!"; "Example of a question icon. Would you like to know the
 current document font?"; IconQuestion! | YesNo!)
 IF(vAns = Yes)
 vFont := ?DocumentFont
 vPrompt := "The current document font is " + vFont
 MESSAGEBOX(x; "FONT!"; vPrompt; IconInformation!)
 GO(LoopToStart)
 ELSE
 GO(LoopToStart)
 ENDIF

LABEL(Stop)
 MESSAGEBOX(z; "STOP!"; "Example of a stop icon"; IconStop!)
 GO(LoopToStart)
```

```

LABEL(LoopToStart)
 MESSAGEBOX(vAns; "Message"; "Would you like to continue?"; IconQuestion! | YesNo!)
 IF(vAns = 7)
 GO(QuitMacro)
 ELSE
 GO(Start)
 ENDIF

LABEL(QuitMacro)
 BEEP
 QUIT

```

---

## See Also

- ♦ PROMPT

---

# MMPLAY

## Purpose

Plays a sound file.

## Syntax

MMPLAY (*Filename*)

## Description

Plays a wave audio file, an Audio-Visual Interleave (AVI) movie file, or a Musical Instrument Digital Interface (MIDI) file, or speaks an ASCII text file (see MMSPEAK). Microsoft Video drivers must be installed to play AVI files. For WAV files, install Windows Audio drivers. MIDI files require Windows MIDI drivers. Speech requires text-to-speech drivers like those shipped with some sound boards.

## Parameters

### *Filename*

Character Expression: The path and name of a sound file. The appropriate driver for each file type must be installed.

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Play .wav file
// Assume Windows sound system and that appropriate driver is installed
// Demonstrate MMPLAY and WAIT commands

No := 7
REPEAT
 MENU(vWav; Digit;; {"Siren"; "Crickets"; "Snoring"; "Thunder"; "Dog bark"; "Elephant"; "Quit"})
 SWITCH(vWav)
 CASEOF 1: PlaySound("siren.wav"; 115)
 CASEOF 2: PlaySound("crickets.wav"; 50)
 CASEOF 3: PlaySound("snoring.wav"; 30)
 CASEOF 4: PlaySound("thunder.wav"; 30)
 CASEOF 5: PlaySound("dogbark.wav"; 15)
 CASEOF 6: PlaySound("elephant.wav"; 15)
 CASEOF 7: QUIT

```

```

ENDSWITCH
MESSAGEBOX(vStatus; "MMPLAY"; "Would you like to hear another sound?"; IconQuestion! |
YesNo!)
UNTIL(vStatus = No)

PROCEDURE PlaySound(vSound; vWait)
 MMPLAY("c:\sndsys\sounds\" + vSound)
 WAIT(vWait)
ENDPROC

```

---

## See Also

♦ MMSPEAK ♦ MMSTOPSPEECH ♦ MMSPEAKCLIPBOARD

---

# MMSPEAK

---

## Purpose

Speaks an ASCII file.

---

## Syntax

MMSPEAK (*Filename*)

---

## Parameters

### *Filename*

Character Expression: The path and name of an ASCII file. A text-to-speech driver must be installed.

---

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Speak an ASCII text file
// A text-to-speech driver must be installed
// Demonstrate MMSPEAK, MMSPEAKCLIPBOARD, MMSTOPSPEECH commands

REPEAT
 MENU(vChoice; Digit;;; {"File"; "Clipboard"; "Quit"})
 SWITCH(vChoice)
 CASEOF 1:
 GETSTRING(vStr; "Enter the path and name of a file to speak."; "ASCII TEXT FILE")
 IF(vStr = "")
 BREAK
 ELSE
 MMSPEAK(vStr)
 CancelMessage()
 ENDIF
 CASEOF 2:
 MMSPEAKCLIPBOARD
 CancelMessage()
 ENDSWITCH
UNTIL(vChoice = 3)
QUIT

```

```

PROCEDURE CancelMessage()
 vCancel := 2
 MESSAGEBOX(vStatus; "SPEAK FILE"; "Choose Cancel any time to stop speech"; IconStop! |
 RetryCancel!)
 IF(vStatus = vCancel)
 MMSTOPSPEECH
 ENDIF
ENDPROC

```

---

**See Also**

• MMPLAY • MMSTOPSPEECH • MMSPEAKCLIPBOARD

---

## MMSPEAKCLIPBOARD

|                    |                                              |
|--------------------|----------------------------------------------|
| <b>Purpose</b>     | Speaks an ASCII file saved to the Clipboard. |
| <b>Syntax</b>      | MMSPEAKCLIPBOARD ()                          |
| <b>Description</b> | A text-to-speech driver must be installed.   |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> APPLICATION(A1; "WordPerfect"; Default; "US") // Speak an ASCII text file // A text-to-speech driver must be installed // Demonstrate MMSPEAK, MMSPEAKCLIPBOARD, MMSTOPSPEECH commands  REPEAT     MENU(vChoice; Digit;;; {"File"; "Clipboard"; "Quit"})     SWITCH(vChoice)         CASEOF 1:             GETSTRING(vStr; "Enter the path and name of a file to speak."; "ASCII TEXT FILE")             IF(vStr = "")                 BREAK             ELSE                 MMSPEAK(vStr)                 CancelMessage()             ENDIF         CASEOF 2:             MMSPEAKCLIPBOARD             CancelMessage()         ENDSWITCH     UNTIL(vChoice = 3) QUIT </pre> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

PROCEDURE CancelMessage()
 vCancel := 2
 MESSAGEBOX(vStatus; "SPEAK FILE"; "Choose Cancel any time to stop speech"; IconStop! |
 RetryCancel!)
 IF(vStatus = vCancel)
 MMSTOPSPEECH
 ENDIF
ENDPROC

```

---

**See Also**

♦ MMPLAY ♦ MMSPEAK ♦ MMSTOPSPEECH

---

## MMSTOPSPEECH

---

|                    |                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Stops a speech (text) file started with MMPLAY (ASCII file), MMSPEAK, or MMSPEAKCLIPBOARD.                                  |
| <b>Syntax</b>      | MMSTOPSPEECH ()                                                                                                             |
| <b>Description</b> | After a text file begins to speak, control immediately returns to the macro. Use MMSTOPSPEECH to stop speaking a text file. |

---

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Speak an ASCII text file
// A text-to-speech driver must be installed
// Demonstrate MMSPEAK, MMSPEAKCLIPBOARD, MMSTOPSPEECH commands

REPEAT
 MENU(vChoice; Digit;;; {"File"; "Clipboard"; "Quit"})
 SWITCH(vChoice)
 CASEOF 1:
 GETSTRING(vStr; "Enter the path and name of a file to speak."; "ASCII TEXT FILE")
 IF(vStr = "")
 BREAK
 ELSE
 MMSPEAK(vStr)
 CancelMessage()
 ENDIF
 CASEOF 2:
 MMSPEAKCLIPBOARD
 CancelMessage()
 ENDSWITCH
UNTIL(vChoice = 3)
QUIT

PROCEDURE CancelMessage()
 vCancel := 2
 MESSAGEBOX(vStatus; "SPEAK FILE"; "Choose Cancel any time to stop speech"; IconStop! |
 RetryCancel!)

```

```
IF(vStatus = vCancel)
 MMSTOPSPEECH
ENDIF
ENDPROC
```

---

**See Also** ♦ MMPLAY ♦ MMSPEAK ♦ MMSPEAKCLIPBOARD

---

## NEST

---

**Purpose** Calls (starts) a nested macro.

---

**Syntax** NEST (*MacroFileName*)

---

**Description** This command is included for DOS compatibility. The Windows equivalent is RUN.

A nested macro starts immediately when called. When a nested macro ends, the first statement following NEST is executed. A macro must be compiled before it is called by NEST.

---

### Parameters

*MacroFileName* Character Expression: The path and name of a compiled macro.

---

**Example** See RUN.

---

**See Also** ♦ CHAIN ♦ RUN

---

## NEWDEFAULT

---

**Purpose** Specifies a macro's new default application.

---

**Syntax** NEWDEFAULT (*ProductPrefix*)

---

**Description** The compiler verifies that each product command is valid for the default application. Invalid product commands produce a syntax error during compilation.

Product commands to the default application do not require a product prefix. A product command to a non-default application, without a product prefix, creates a compile-time syntax error.

---

## Parameters

*ProductPrefix*

Character Expression: A unique identifier that matches the *ProductPrefix* parameter of an APPLICATION statement.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
APPLICATION(A2; ... identify another application ...)

// WordPerfect is the default application
// AboutDlg does not require a product prefix
// Demonstrate NEWDEFAULT command

AboutDlg()

NEWDEFAULT(A2)

// WordPerfect is not the default application
// AboutDlg requires a product prefix (A1)
A1.AboutDlg()
```

---

## See Also

♦ APPLICATION ♦ ENDAPP

---

---

# NEXT

---

## Purpose

Advances a loop iteration before the end of the loop.

---

## Syntax

NEXT

---

## Description

Use NEXT inside a conditional statement such as IF or SWITCH in a loop to advance a loop iteration when a specified condition is met. Statements after NEXT are ignored.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test Not Found condition
// Demonstrate NEXT command

NOTFOUND(On!)
ONNOTFOUND CALL(NotFoundTest)
No := 7
REPEAT
 GETSTRING(vStr; "Enter a string to search for and press <Enter>."; "ONNOTFOUND CALL
 EXAMPLE")
 IF(vStr = "")
 NEXT
 ENDIF
 vError := False
 SearchString(vStr)
 SearchNext(Extended!)
```



```

IF(vError = False)
 vMessage := "Success! Do you want to try again?"
ELSE
 vMessage := "Not Found condition. Do you want to try again?"
ENDIF
MESSAGEBOX(vStatus; "Search"; vMessage; IconInformation! | YesNo!)
UNTIL(vStatus = No)
QUIT

LABEL(NotFoundTest)
BEEP
MESSAGEBOX(vError; "ERROR"; """" + vStr + """" not found"; IconExclamation!)
vError := True
RETURN // return to IF statement following SearchNext command

```

---

## See Also

♦ BREAK ♦ FOR ♦ FOREACH ♦ FORNEXT ♦ REPEAT ♦ WHILE

---

# NOTFOUND

---

## Description

Determines how a macro responds to a Not Found condition.

---

## Syntax

NOTFOUND (*State*)

---

## Description

A Not Found condition is generally the result of a failed search. Create a Not Found condition with ASSERT(NotFoundCondition!).

---

## Parameters

### *State*

Enumeration: Specifies the Not Found state. The default is NOTFOUND(On!). The enumerations are:

- Off! Overrides a Not Found condition.
- On! Stops a macro unless preceded by ONNOTFOUND, which directs macro execution to a specified LABEL.

---

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// LABEL(NotFoundMsg) executes with NOTFOUND(On!)
// Demonstrate NOTFOUND command

ONNOTFOUND(NotFoundMsg)

REPEAT
 MENU(vChoice; Digit;;; {"NOTFOUND(Off!); "NOTFOUND(On!); "Quit"})
 CASE CALL(vChoice; {1; OffMsg; 2; OnMsg; 3; QuitMacro}; QuitMacro)
UNTIL(vChoice = 3)

```

```

LABEL(OffMsg)
 NOTFOUND(Off!)
 ASSERT(NotFoundCondition!) // ignore Not Found condition
 MESSAGEBOX(x; "NOTFOUND(Off!); "Not Found condition ignored"; IconInformation!)
 RETURN

```

```

LABEL(OnMsg)
 NOTFOUND(On!)
 ASSERT(NotFoundCondition!) // LABEL(NotFoundMsg) executed
 RETURN

```

```

LABEL(NotFoundMsg)
 BEEP
 MESSAGEBOX(x; "NOTFOUND CONDITION"; "A Not Found condition causes
 LABEL(NotFoundMsg) to execute"; IconInformation!)
 RETURN

```

```

LABEL(QuitMacro)
 QUIT

```

---

## See Also

♦ ASSERT ♦ CANCEL ♦ ERROR ♦ LABEL ♦ ONNOTFOUND

---

# NTOC

**Purpose** Converts a decimal value to its character equivalent.

**Syntax** ReturnValue := NTOC (*CharSet*; *CharValue*)

**Returns** The character equivalent of a decimal value.

## Parameters

*CharSet* Numeric Expression (optional): The number of a WordPerfect character set. If this parameter is not used, the default is 0 (ASCII set).

*CharValue* Numeric Expression: The number of a character in a WordPerfect character set.

---

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Convert integer to Character equivalent
// NTOC(32) returns "space" character
// Demonstrate NTOC command

Display(On!)

```



---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Most type conversions are handled automatically
// The following type statements output identical character strings
// Demonstrate NUMSTR command

HdReturn := NTOC(0F90Ah)
PosDocBottom
TotalPgNمبر := ?Page
TotalPgString := NUMSTR(TotalPgNمبر)

MESSAGEBOX(x; "AUTOMATIC TYPE CONVERSION"; "Variable TotalPgNمبر: There are " +
TotalPgNمبر + " pages in this document." + HdReturn + HdReturn + "Variable TotalPgString: There are "
+ TotalPgString + " pages in this document." + HdReturn; IconInformation!)
```

---

**See Also**

- ♦ STRNUM

---

## ONCANCEL

---

**Purpose**

Executes a LABEL statement when a Cancel condition occurs.  
IMPORTANT: A Cancel condition stops a macro unless preceded by ONCANCEL.

---

**Syntax**

ONCANCEL (*Label*)

---

**Description**

A macro with more than one ONCANCEL executes the last ONCANCEL before a Cancel condition occurs.

---

**Parameters***Label*

Label: A LABEL to execute when a Cancel condition occurs.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate ONCANCEL command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
MENU(vPick; Digit!;;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
SWITCH(vPick)
CASEOF 1: ASSERT(CancelCondition!)
CASEOF 2: ASSERT(ErrorCondition!)
CASEOF 3: ASSERT(NotFoundCondition!)
CASEOF 4: QUIT
DEFAULT: GO(Condition)
ENDSWITCH
```

```

LABEL(Condition)
 SWITCH(ErrorNumber)
 CASEOF 1: vMsg := "1 (Cancel condition)"
 CASEOF 2: vMsg := "2 (Error condition)"
 CASEOF 7: vMsg := "3 (Not Found condition)"
 DEFAULT: vMsg := "?"
 ENDSWITCH
MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
GO(Start)

```

---

**See Also**

♦ ASSERT ♦ LABEL ♦ ONCANCEL CALL ♦ ONERROR ♦ ONNOTFOUND

---

## ONCANCEL CALL

---

**Purpose** Calls a LABEL-RETURN statement when a Cancel condition occurs. RETURN directs macro execution to the first statement after the Cancel condition.

**IMPORTANT:** A Cancel condition stops a macro unless preceded by ONCANCEL CALL.

---

**Syntax** ONCANCEL CALL (*Label*)

---

**Description** A macro with more than one ONCANCEL CALL executes the last ONCANCEL CALL before a Cancel condition occurs.

---

**Parameters**

*Label* Label: A LABEL to call when a Cancel condition occurs.

---

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input for Y or N
// For Cancel condition: chose Cancel, click Close, press Alt+F4, or double-click system menu box
// Demonstrate ONCANCEL CALL command

```

```

CANCEL(On!)
ONCANCEL CALL(CancelMessage)
PROCEDURE Message(vPrompt; vlcon)
 IF(vlcon = "!")
 BEEP
 MESSAGEBOX(y; "Message Box"; vPrompt; IconExclamation!)
 ELSE
 MESSAGEBOX(y; "Message Box"; vPrompt; IconInformation!)
 ENDIF
ENDPROC

```

```

vStr := ""
REPEAT
 GETSTRING(vStr, "Press one character: ""Y"" for Yes or ""N"" for no"; "GETSTRING EXAMPLE";
 50)
 IF(vStr = "")
 vStr := "<Enter>"
 ENDIF

 vTest := TOUPPER(vStr)
 IF(CHARLEN(vStr) > 1)
 CALL Message("You typed or pressed "" + vStr + """"; "!")
 ELSE
 IF((vTest = "Y") OR (vTest = "N"))
 CALL Message("Well done - You pressed " + vTest; "i")
 ELSE
 CALL Message("You pressed "" + vStr + """"; "!")
 ENDIF
 ENDIF
UNTIL(((vTest = "Y") OR (vTest = "N")) AND NOT(CHARLEN(vStr) > 1))
QUIT

LABEL(CancelMessage)
vStr := "a Cancel control"
RETURN

```

---

## See Also

♦ ASSERT ♦ LABEL ♦ ONCANCEL ♦ ONCANCEL CALL ♦ ONDDEADVISE CALL ♦ ONERROR ♦ ONERROR CALL ♦ ONNOTFOUND ♦ ONNOTFOUND CALL ♦ RETURN

---

# ONDDEADVISE CALL

---

## Purpose

Establishes a link with an application and tests for changes in the value of the DDEREQUEST *ItemName* parameter.

---

## Syntax

ONDDEADVISE CALL (*MacroVar*; *ConversationID*; *ItemName*; *Label*)

---

## Parameters

*MacroVar*

Variable: 0 is returned in this variable if the ONDDEADVISE CALL link is successfully established, or a number greater than 0 if a link is not established.

*ConversationID*

Numeric Expression: The value of the DDEREQUEST *ConversationID* parameter.

*ItemName*

Character Expression: The value of the DDEREQUEST *ItemName* parameter.

*Label* Label: The LABEL to call when the value of *ItemName* changes.

**See Also** ♦ DDEINITIATE ♦ DDEREQUEST ♦ ONCANCEL CALL ♦ ONERROR CALL  
♦ ONNOTFOUND CALL

---

## ONERROR

---

**Purpose** Executes a LABEL statement when an Error condition occurs.  
IMPORTANT: An Error condition stops a macro unless preceded by ONERROR.

---

**Syntax** ONERROR (*Label*)

---

**Description** A macro with more than one ONERROR statement executes the last ONERROR before an Error condition occurs.

---

### Parameters

*Label* Label: A LABEL to execute when an Error condition occurs.

---

### Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate ONERROR command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
 MENU(vPick; Digit;;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
 SWITCH(vPick)
 CASEOF 1: ASSERT(CancelCondition!)
 CASEOF 2: ASSERT(ErrorCondition!)
 CASEOF 3: ASSERT(NotFoundCondition!)
 CASEOF 4: QUIT
 DEFAULT: GO(Condition)
 ENDSWITCH

LABEL(Condition)
 SWITCH(ErrorNumber)
 CASEOF 1: vMsg := "1 (Cancel condition)"
 CASEOF 2: vMsg := "2 (Error condition)"
 CASEOF 7: vMsg := "3 (Not Found condition)"
 DEFAULT: vMsg := "?"
 ENDSWITCH
 MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
GO(Start)
```

---

**See Also**

♦ ASSERT ♦ LABEL ♦ ONCANCEL ♦ ONERROR CALL ♦ ONNOTFOUND

---

## ONERROR CALL

---

**Purpose**

Calls a LABEL-RETURN statement when an Error condition occurs. RETURN directs macro execution to the first statement after the Error condition.

IMPORTANT: An Error condition stops a macro unless preceded by ONERROR CALL.

---

**Syntax**

ONERROR CALL (*Label*)

---

**Description**

A macro with more than one ONERROR CALL statement executes the last ONERROR CALL before an Error condition occurs.

---

**Parameters***Label*

Label: A LABEL to call when an Error condition occurs.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test Error condition
// Demonstrate ONERROR CALL command

ONERROR CALL(ErrorTest)
No := 7
REPEAT
 GETNUMBER(vNbr; "Enter 3 and press <Enter>."; "ONERROR CALL EXAMPLE")
 IF(vNbr = 3)
 MESSAGEBOX(vStatus; "YES!"; "You entered 3 as requested. Do you want to try again?";
 IconInformation! | YesNo!)
 ELSE
 ASSERT(ErrorCondition!)
 ENDIF
UNTIL(vStatus = No)
QUIT

LABEL(ErrorTest)
BEEP
MESSAGEBOX(vStatus; "ERROR"; "You didn't enter 3 as requested"; IconExclamation!)
RETURN
```

---

**See Also**

♦ ASSERT ♦ LABEL ♦ ONCANCEL ♦ ONCANCEL CALL ♦ ONERROR  
♦ ONNOTFOUND ♦ ONNOTFOUND CALL ♦ RETURN

---



---

# ONNOTFOUND

---

## Purpose

Executes a LABEL statement when a Not Found condition occurs.  
IMPORTANT: A Not Found condition stops a macro unless preceded by ONNOTFOUND.

## Syntax

ONNOTFOUND (*Label*)

## Description

A macro with more than one ONNOTFOUND statement executes the last ONNOTFOUND before a Not Found condition occurs.

## Parameters

*Label* Label: A LABEL to execute when a Not Found condition occurs.

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate ONNOTFOUND command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
 MENU(vPick; Digit;;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
 SWITCH(vPick)
 CASEOF 1: ASSERT(CancelCondition!)
 CASEOF 2: ASSERT(ErrorCondition!)
 CASEOF 3: ASSERT(NotFoundCondition!)
 CASEOF 4: QUIT
 DEFAULT: GO(Condition)
 ENDSWITCH

LABEL(Condition)
 SWITCH(ErrorNumber)
 CASEOF 1: vMsg := "1 (Cancel condition)"
 CASEOF 2: vMsg := "2 (Error condition)"
 CASEOF 7: vMsg := "3 (Not Found condition)"
 DEFAULT: vMsg := "?"
 ENDSWITCH
 MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
 GO(Start)
```

---

## See Also

♦ ASSERT ♦ LABEL ♦ ONCANCEL ♦ ONERROR ♦ ONNOTFOUND CALL

---

# ONNOTFOUND CALL

---

**Purpose** Calls a LABEL-RETURN statement when a Not Found condition occurs. RETURN directs macro execution to the first statement after the Not Found condition.

IMPORTANT: A Not Found condition stops a macro unless preceded by ONNOTFOUND CALL.

---

**Syntax** ONNOTFOUND CALL (*Label*)

---

**Description** A macro with more than one ONNOTFOUND CALL statement executes the last ONNOTFOUND CALL before a Not Found condition occurs.

---

## Parameters

*Label* Label: A LABEL to call when a Not Found condition occurs.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test Not Found condition
// Demonstrate ONNOTFOUND CALL command

NOTFOUND(On!)
ONNOTFOUND CALL(NotFoundTest)
No := 7
REPEAT
 GETSTRING(vStr; "Enter a string to search for and press <Enter>: "; "ONNOTFOUND CALL
 EXAMPLE")
 IF(vStr = "")
 NEXT
 ENDIF
 vError := False
 SearchString(vStr)
 SearchNext(Extended!)
 IF(vError = False)
 vMessage := "Success! Do you want to try again?"
 ELSE
 vMessage := "Not Found condition. Do you want to try again?"
 ENDIF
 MESSAGEBOX(vStatus; "Search"; vMessage; IconInformation; ! YesNo!)
UNTIL(vStatus = No)
QUIT

LABEL(NotFoundTest)
BEEP
MESSAGEBOX(vError; "ERROR"; "" + vStr + "" not found"; IconExclamation!)
vError := True
RETURN // return to IF statement following SearchNext command
```

---

**See Also**

♦ ASSERT ♦ LABEL ♦ ONCANCEL ♦ ONCANCEL CALL ♦ ONERROR  
♦ ONERROR CALL ♦ ONNOTFOUND ♦ RETURN

---

# PAUSE

---

**Purpose**

Pauses a macro, returning computer control to the user.

---

**Syntax**

PAUSE

---

**Description**

To resume macro execution, press Enter, or choose Tools, Macro, Pause. To stop a macro, choose Tools, Macro, Stop. When PAUSE follows a PROMPT statement, choose OK to resume macro execution or Cancel to create a Cancel condition.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate PAUSE command

ONCANCEL(EndLoop)
Display(On!)
InhibitInput(Off!)
InformationIcon := 4
x := 1
FileNew
PROMPT("PAUSE EXAMPLE"; "Choose OK to dismiss prompt and continue loop, or Cancel to end
loop."; InformationIcon)
REPEAT
 Type(x + NTOC(32))
 IF(x MOD 10 = 0)
 PAUSE
 Type(" - Press OK to continue or Esc to end")
 HardReturn
 ENDIF
 x := x + 1
UNTIL(x = 0)

LABEL(EndLoop)
QUIT
```

---

**See Also**

♦ CANCEL ♦ ONCANCEL ♦ PROMPT ♦ PauseSet

---

---

# PERSIST

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Declares Shared Code variables and arrays, and assigns them to the persist variable table.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Syntax</b>      | PERSIST ( <i>VariableName</i> ; <i>VariableName</i> ; ... <i>VariableName</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Description</b> | PERSIST variables are available to any Shared Code application while Shared Code is running.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Parameters</b>  | <p><i>VariableName</i> Variable: One or more user-defined variables separated by a semicolon. Variables must begin with a letter and can include any other combination of letters or numbers. Separate multiple variables with a semicolon. Parentheses are optional. For example,</p> <pre>PERSIST <i>VariableName</i>; <i>VariableName</i> PERSIST(<i>VariableName</i>; <i>VariableName</i>)</pre> <p>are both valid statements. The next example declares persist arrays with 5 elements and 15 elements:</p> <pre>PERSIST FirstArray[5]; NextArray[15]</pre> |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre>APPLICATION(A1; "WordPerfect"; Default; "US") // Declare and test variable types // Demonstrate BREAK, EXIST, LOCAL, GLOBAL, and PERSIST commands  DISCARD var5; var1; var7 LOCAL var2; var9; var4 GLOBAL var3; var10; var6 PERSIST var5; var1; var7 ASSIGN(var8; "")  FOR(x; 1; x &lt; 11; x + 1)   ASSIGN(INDIRECT("var" + x); x)   MESSAGEBOX(vStatus; "VAR" + x; "Continue to next variable?"; IconQuestion!   YesNo!)   IF(vStatus = 7)     BREAK   ENDIF ENDFOR  FOR(z; 1; z &lt; (x + 1); z + 1)   SWITCH(EXISTS(INDIRECT("var" + z)))     CASEOF 1: Msg("Local"; z)     CASEOF 2: Msg("Global"; z)     CASEOF 3: Msg("Persist"; z)   ENDSWITCH ENDFOR  FUNCTION Msg(w; y)   MESSAGEBOX(z; "VARIABLE TYPES"; "var" + y + " = " + w) ENDFUNC</pre> |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

# PERSISTALL

---

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Assigns a macro's variables and arrays to the Persist variable table.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Syntax</b>      | PERSISTALL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Description</b> | <p>PERSIST variables are available to any Shared Code application while Shared Code is running.</p> <p>Only variables assigned values after PERSISTALL are Persist variables. Variables assigned values before PERSISTALL, and variables declared LOCAL or GLOBAL, are not Persist variables. For example, only var6 and var7 are Persist variables:</p> <pre> APPLICATION(A1; "WordPerfect"; Default; "US") LOCAL var1; var2 ASSIGN(var3; 0) GLOBAL var4 PERSIST var7 ASSIGN(var7; "Persist Variable")  PERSISTALL  LOCAL var5 ASSIGN(var4; 2) ASSIGN(var6; "Persist Variable") </pre> |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> APPLICATION(A1; "WordPerfect"; Default; "US") // Name of macro: Persist.wcm // vNbr is a Persist variable // Demonstrate PERSISTALL command  MESSAGEBOX(x; "PERSISTALL EXAMPLE"; "The value of variable vNbr is " + vNbr; IconInformation!) ASSIGN(vNbr; 999) RETURN  APPLICATION(A1; "WordPerfect"; Default; "US") // Run PERSIST.WCM macro // Demonstrate PERSISTALL command  LOCAL vStatus; vCancel vStatus := 4 vCancel := 2  PERSISTALL // vNbr Persist variable </pre> |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

REPEAT
 IF(EXISTS(vNnbr))
 DISCARD vNnbr
 ENDIF
 GETNUMBER(vNnbr; "Enter a number: "; "PERSISTALL EXAMPLE")
 vMacrosPath := ?PathMacros
 RUN(vMacrosPath + "persist.wcm")
 MESSAGEBOX(vStatus; "PERSISTALL EXAMPLE"; "After running PERSIST.WCM, the value of
 variable vNnbr is " + vNnbr; IconInformation! | RetryCancel!)
UNTIL(vStatus = vCancel)

```

---

**See Also**

♦ DISCARD ♦ GLOBAL ♦ LOCAL ♦ PERSIST

---

## PROCEDURE

---

**Purpose**

Identifies a macro subroutine that can receive one or more values from a calling statement (see Calling Statements).

---

**Syntax**

The general form of a PROCEDURE statement is,

```

PROCEDURE Name (Parameter; Parameter; ... Parameter)
...statement block...
ENDPROC

```

---

**Description**

Procedures contain one or more statements that execute when the procedure is called. Unlike LABEL subroutines, procedures do not execute unless called. A calling statement consists of the procedure's name, and can have one or more parameters that contain values passed to the procedure. RETURN or ENDPROC direct macro execution to the statement that follows the procedure's caller.

**Address Mode**

To change the value of a variable passed to a procedure, precede the calling statement parameter and its corresponding procedure parameter with an ampersand (&). In the following example, the value of x is 5 after calling Test:

```

x := 5
Test(x) // calling statement

```

```

PROCEDURE Test(z)
z := z + 5
ENDPROC

```

In the next example, the value of x is 10 after calling Test:

```

x := 5
Test(&x) // calling statement

```

```
PROCEDURE Test(&z)
 z := z + 5
ENDPROC
```

The ampersand before variable `x` means the variable's address (location in memory) is passed to the procedure, not the variable's value. Changes made at the address of `x` are made to the contents of `x`.

---

## Parameters

|                  |                                                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Name</i>      | Label: The name of the procedure. It begins with a letter and consists of one or more letters or numbers.                                                                                                                                                                                                                   |
| <i>Parameter</i> | Variable: Receives a value from a calling statement (see <i>Calling Statements in Macro Control Statements</i> ). If an ampersand precedes the calling statement variable, an ampersand must precede the corresponding procedure variable (see <i>Address Mode</i> above). Multiple variables are separated by a semicolon. |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input
// Remember: procedures do not execute unless called
// Demonstrate PROCEDURE command

PROCEDURE ErrorMsg(x)
 BEEP BEEP BEEP
 IF(x < 1)
 vMessage := x + " is less than 1"
 ELSE
 vMessage := x + " is greater than 5"
 ENDIF
 MESSAGEBOX(vStatus; "OUT-OF-RANGE ERROR"; vMessage + " - please try again";
 IconExclamation!)
ENDPROC

REPEAT
 GETNUMBER(vNum; "Enter a number from 1 to 5"; "GETNUMBER EXAMPLE")
 IF((vNum < 1) OR (vNum > 5))
 CALL ErrorMsg(vNum)
 ELSE
 MESSAGEBOX(vStatus; "YES!"; "You entered " + vNum; IconInformation!)
 ENDIF
UNTIL((vNum > 0) AND (vNum < 6))
```

---

## See Also

♦ CALL ♦ FUNCTION ♦ LABEL ♦ RETURN ♦ USE

---

# PROMPT

---

## Purpose

Displays a message box with OK and Cancel button controls.

IMPORTANT: You can display only one message box at a time. Calling a second message box replaces the first.

---

## Syntax

The general form of a PROMPT statement is,

```
PROMPT (Title; Prompt; Icon; HorizPosition; VertiPosition)
PAUSE
```

A PROMPT statement can have another form,

```
PROMPT (Title; Prompt; Icon; HorizPosition; VertiPosition)
```

*...other statements...*

```
ENDPROMPT
```

---

## Description

PAUSE following PROMPT displays a PROMPT message box until the user selects OK or Cancel. OK removes the prompt and resumes macro execution. Cancel removes the prompt and creates a Cancel condition (see ONCANCEL).

If PROMPT is not followed by PAUSE, the message box is displayed until an ENDPROMPT occurs or the macro ends.

---

## Parameters

*Title*

Character Expression: The text displayed in the title bar.

*Prompt*

Character Expression (optional): The text displayed in the prompt box. If not used, the title is displayed as the prompt and "WordPerfect Macro Facility" as the title.

*Icon*

Numeric Expression (optional): The number of an icon displayed to the left of *Prompt*. The default value is zero. The values are,

- 0 (no icon)
- 1 (stop sign)
- 2 (question mark)
- 3 (exclamation point)
- 4 (information icon)

*HorizPosition*

Numeric Expression (optional): The number of pixels from the left side of the main window to the left side of the message box. If you omit this parameter, the prompt is centered horizontally in the main window.

*VertiPosition*

Numeric Expression (optional): The number of pixels from the top of the main window to the top of the message box. If you omit this parameter, the prompt is centered vertically in the main window.

---



---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate PROMPT command

ONCANCEL(EndLoop)
Display(On!)
InhibitInput(Off!)
InformationIcon := 4
x := 1
FileNew
PROMPT("PAUSE EXAMPLE"; "Choose OK to dismiss prompt and continue loop, or Cancel to end
loop."; InformationIcon)
REPEAT
 Type(x + NTOC(32))
 IF(x MOD 10 = 0)
 PAUSE
 Type(" - Press OK to continue or Esc to end")
 HardReturn
 ENDIF
 x := x + 1
UNTIL(x = 0)

LABEL(EndLoop)
QUIT
```

---

**See Also**

♦ CANCEL ♦ MESSAGEBOX ♦ ONCANCEL ♦ PAUSE

---

# QUIT

---

**Purpose**

Ends a macro.

---

**Syntax**

QUIT

---

**Description**

CHAIN does not execute if QUIT occurs before CHAIN.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input for Y or N
// For Cancel condition: chose Cancel, click Close, press Alt+F4, or double-click system menu box
// Demonstrate QUIT command

CANCEL(On!)
ONCANCEL CALL(CancelMessage)
PROCEDURE Message(vPrompt; vIcon)
 IF(vIcon = "!")
 BEEP
 MESSAGEBOX(y; "Message Box"; vPrompt; IconExclamation!)
 ELSE
```

```

 MESSAGEBOX(y; "Message Box"; vPrompt; IconInformation!)
 ENDIF
ENDPROC

vStr := ""
REPEAT
 GETSTRING(vStr; "Press one character: ""Y"" for Yes or ""N"" for no"; "GETSTRING EXAMPLE";
 50)
 IF(vStr = "")
 vStr := "<Enter>"
 ENDIF

 vTest := TOUPPER(vStr)
 IF(CHARLEN(vStr) > 1)
 CALL Message("You typed or pressed "" + vStr + """"; "!")
 ELSE
 IF((vTest = "Y") OR (vTest = "N"))
 CALL Message("Well done - You pressed " + vTest; "i")
 ELSE
 CALL Message("You pressed "" + vStr + """"; "!")
 ENDIF
 ENDIF
UNTIL(((vTest = "Y") OR (vTest = "N")) AND NOT(CHARLEN(vStr) > 1))
QUIT

LABEL(CancelMessage)
vStr := "a Cancel control"
RETURN

```

---

## See Also

- ◆ CHAIN

---

# REPEAT-UNTIL

---

## Purpose

A loop statement that executes until the expression at the bottom of the loop is true. See Loop Statements.

---

## Syntax

The general form of a REPEAT statement is,

```

REPEAT
...statement block...
UNTIL (Test)

```

---

## Description

The loop executes at least once, because it is not tested until the bottom of the loop. When *Test* is true, the first statement after UNTIL is executed.

---

## Parameters

*Test* Relational Expression: Evaluates to true or false.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input
// Remember: procedures do not execute unless called
// Demonstrate REPEAT command

PROCEDURE ErrorMsg(x)
 BEEP BEEP BEEP
 IF(x < 1)
 vMessage := x + " is less than 1"
 ELSE
 vMessage := x + " is greater than 5"
 ENDIF
 MESSAGEBOX(vStatus; "OUT-OF-RANGE ERROR"; vMessage + " - please try again";
 IconExclamation!)
ENDPROC

REPEAT
 GETNUMBER(vNum; "Enter a number from 1 to 5"; "GETNUMBER EXAMPLE")
 IF((vNum < 1) OR (vNum > 5))
 CALL ErrorMsg(vNum)
 ELSE
 MESSAGEBOX(vStatus; "YES!"; "You entered " + vNum; IconInformation!)
 ENDIF
UNTIL((vNum > 0) AND (vNum < 6))
```

---

## See Also

♦ FOR ♦ FOREACH ♦ FORNEXT ♦ WHILE

---

---

# RETURN

**Purpose** Ends LABEL, FUNCTION, and PROCEDURE subroutines, or a macro (see RUN), and then directs macro execution to the statement that follows the subroutine or macro's caller, or creates a Cancel, Error, or Not Found condition.

---

**Syntax** RETURN (*Condition; Value*)

---

**Description** RETURN generally ends a LABEL statement called by statements such as CALL or CASE CALL.

```
CALL(StartMacro)
CALL(QuitMacro)

LABEL(StartMacro)
... statement block...
RETURN // directs macro execution to CALL(QuitMacro)
```

```
LABEL(QuitMacro)
QUIT
```

If there is no caller to return to, and the macro containing RETURN is nested (called by another macro), RETURN directs macro execution to the statement that follows the macro's caller (see RUN). RETURN ends a macro if there is no caller to return to, and the macro containing the RETURN statement is not nested.

---

## Parameters

### *Condition*

Enumeration (optional): Creates a Cancel, Error, or Not Found condition (see ASSERT). *Condition* stops a macro unless preceded by ONCANCEL, ONERROR, or ONNOTFOUND, which direct macro execution to a specified LABEL. *Condition* has no effect when preceded by CANCEL(Off!), ERROR(Off!), or NOTFOUND(Off!).

The *Condition* return types are,

|                    |                                              |
|--------------------|----------------------------------------------|
| CancelCondition!   | Stops a macro unless preceded by ONCANCEL.   |
| ErrorCondition!    | Stops a macro unless preceded by ONERROR.    |
| NotFoundCondition! | Stops a macro unless preceded by ONNOTFOUND. |

### *Value*

Any (optional): Returns the result of a function operation (see FUNCTION).

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Execute Windows Calculator, Calendar, and Cardfile
// Demonstrate RETURN command

ASSIGN(hWP; APPLOCATE(""))
REPEAT
 MENU(vChoice; Digit;; {"Calculator"; "Calendar"; "Card File"; "Quit"})
 CASE CALL(vChoice; {1; Program; 2; Program; 3; Program; 4; QuitMacro; QuitMacro})
UNTIL(vChoice = 4)

LABEL(Program)
 SWITCH(vChoice)
 CASEOF 1: vProgram := "c:\windows\calc.exe"
 CASEOF 2: vProgram := "c:\windows\calendar.exe"
 CASEOF 3: vProgram := "c:\windows\cardfile.exe"
 ENDSWITCH
 APPEXECUTE(vProgram)
 RETURN

LABEL(QuitMacro)
 MESSAGEBOX(vStatus; "EXIT"; "Are you finished?"; IconQuestion! | YesNo!)
 IF(vStatus = 6)
 APPACTIVATE(hWP)
 QUIT
 ELSE
 vChoice := ""
 RETURN
 ENDIF
```

---

**See Also**

♦ ASSERT ♦ CALL ♦ CASE CALL ♦ FUNCTION ♦ LABEL ♦ PROCEDURE  
♦ ONCANCEL ♦ ONERROR ♦ ONNOTFOUND ♦ RUN

---

---

**RUN**

---

**Purpose** Calls (starts) a nested macro.

---

**Syntax** RUN (*MacroFile*)

---

**Description** A nested macro starts immediately when it is called. When a nested macro ends, control returns to the calling macro. If QUIT ends a nested macro, control does not return to the calling macro. A macro must be compiled before it is called by RUN.

---

**Parameters**

*MacroFile* Character Expression: The path and name of a compiled macro.

---

**Example**

```
// There are four macros in this example
// Compile separately

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro1.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)
RETURN

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro2.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)
RETURN

APPLICATION(A1; "WordPerfect"; Default; "US")
// macro3.wcm
MESSAGEBOX(vStatus; vTitle; vPrompt; IconInformation!)
RETURN

APPLICATION(A1; "WordPerfect"; Default; "US")
// Calling Macro
// Demonstrate RUN command and GLOBAL variables

GLOBAL vTitle; vPrompt
ASSIGN(vPrompt; "RUN macros can return to their caller")
ASSIGN(vMacrosPath; ?PathMacros)
REPEAT
 MENU(vChoice; Letter;; {"Macro 1"; "Macro 2"; "Macro 3"; "Quit"})
 SWITCH(vChoice)
 CASEOF 1:
 ASSIGN(vTitle; "MACRO ONE")
 RUN(vMacrosPath + "macro1.wcm")
```

```

CASEOF 2:
 ASSIGN(vTitle; "MACRO TWO")
 RUN(vMacrosPath + "macro2.wcm")
CASEOF 3:
 ASSIGN(vTitle; "MACRO THREE")
 RUN(vMacrosPath + "macro3.wcm")
DEFAULT: QuitMacro
ENDSWITCH
UNTIL(vChoice = 4)

LABEL(QuitMacro)
BEEP
MESSAGEBOX(vStatus; "QUIT"; "This example uses global variables"; IconExclamation!)
DISCARD vTitle; vPrompt
QUIT

```

---

## See Also

♦ CHAIN

---

# SENDKEYS WAIT

---

## Purpose

Sends keystrokes to the current application. *WAIT* is optional (see *Description*).  
**IMPORTANT:** Set `InhibitInput(Off!)` for `SENDKEYS` to operate correctly.

---

## Syntax

`SENDKEYS WAIT` (*KeyCode*, *MarkupLanguage*)

---

## Description

Keys such as `Alt` or `F1` must be enclosed in braces to specify a single keystroke. For example,

```
SENDKEYS("{Alt}LLH")
```

sends four keystrokes (`Alt`, `L`, `L`, and `H`) which opens the Line Height dialog box. The following commands are equivalent to the previous example:

```
SENDKEYS("{Alt + L + L + H}")
SENDKEYS("{Alt}{L}{L}{H}")
```

If you combine keystrokes in a single set of braces, you must separate them with a plus operator. Otherwise, the result is unpredictable. Enclosing single character keystrokes in braces is optional.

Assign frequently used keystrokes to a variable. For example,

```
ASSIGN(KS_MarginsDlg; "{Alt}LM")
SENDKEYS(KS_MarginsDlg)
```

opens the Margins dialog box, and is equivalent to the programming command `FormatMarginsDlg`. Do not use variable names that are identical to a macro command name.

A minus operator releases a keystroke. For example,

```
SENDKEYS("{Shift+Del-Del+Ins}")
```

cuts selected text to the clipboard (Shift+Del), then pastes it at the insertion point (Shift+Ins).

With the WAIT modifier, SENDKEYS is processed before the next macro statement. Otherwise, SENDKEYS is processed after the next macro statement. For example,

```
APPLICATION(A1; "WordPerfect"; Default; "US")
Display(On!)
InhibitInput(Off!)
SENDKEYS("C:\WPWIN60\MACROS*.WCM {Enter}")
FileOpenDlg
```

displays the Open File dialog box, enters C:\WPWIN60\MACROS\\*.WCM in the Filename edit box, and presses Enter. The directory changes to C:\WPWIN60\MACROS, and files with a .WCM extension appear in the Filename list box. The same example with the WAIT modifier enters C:\WPWIN60\MACROS\\*.WCM in the document before the Open File dialog box is displayed.

---

## Parameters

### *KeyCode*

Character Expression: Keystrokes to send.

```
{VKnnn} nnn = ANSI character number
{Alt}
{Ctrl}
{Control}
{Shift}
{0} - {9} Digits
{A} - {Z} Alphabet
{F1} - {F16} Function keys
{NumLock}
{NumAdd}
{NumSubtract}
{NumMultiply}
{NumDivide}
{NumDecimal}
{Num0} - {Num9} Numpad numbers
{Left}
{Right}
{Up}
{Dn}
{Down}
{PgDn}
{PageDown}
{PgUp}
{PageUp}
{Bksp}
```

```

{Backspace}
{Break} Cancel
{CapsLock}
{Clear}
{Del}
{Delete}
{End}
{Enter}
{Esc}
{Escape}
{Help} VK Help key
{Home}
{Ins}
{Insert}
{Minus}
{Pause}
{ScrLock}
{ScrollLock}
{PrintScrn}
{PrintScreen}
{Space}
{Tab}
{LeftBrace} {
{RightBrace} }

```

### *MarkupLanguage*

Numeric Expression: Specifies which language interprets the key codes. The values are,

- 1 WPWin 6.0 keystring language.

---

### **Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Save and compile selected text
// To compile and play, use MacroFilePlay command. This example is for demonstration purposes only.
// Demonstrate SENDKEYS WAIT command

IF(?SelectedTextSize > 0)
 SelectSave (Filename: "C:\WPWIN60\MACROS\TEST.WCM")
 hWP := APPLOCATE("*")
 hMF := APPLOCATE("WordPerfect Macro Facility")
 APPACTIVATE(hMF)
 CALL(MacroFacility)
 CALL(ChkForSyntaxError)
 APPACTIVATE(hWP)
 SelectMode(Off!)
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "Select text before playing macro"; IconExclamation!)
ENDIF
QUIT

```



```

LABEL(MacroFacility)
 SENDKEYS WAIT("{Alt+M}{C:c:\wpwin60\macros\test.wcm {Enter}")
 RETURN

LABEL(ChkForSyntaxError)
 hMFSE := APPLOCATE("WordPerfect Macro Facility - Syntax Error")
 IF(hMFSE)
 BEEP
 APPACTIVATE(hMFSE)
 QUIT
 ELSE
 RETURN
 ENDIF

```

---

## SPEED

|                    |                                                                                                                                                                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Slows macro execution.                                                                                                                                                                                                                                             |
| <b>Syntax</b>      | SPEED ( <i>TenthsOfSeconds</i> )                                                                                                                                                                                                                                   |
| <b>Description</b> | <p>Time is measured in tenths of a second. SPEED(0) runs at maximum speed. SPEED(5) delays a macro one-half second between statements. The maximum delay is one minute, or SPEED(600).</p> <pre> SPEED(10) // delay one second between beeps BEEP BEEP BEEP </pre> |

### Parameters

*TenthsOfSeconds*      Numeric Expression: A number from zero to 600. Divide the number by 10 to calculate the number of seconds.

|                |                                                                                                                                                                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | <pre> APPLICATION(A1; "WordPerfect"; Default; "US") // Demonstrate SPEED command  Display(On!) ONCANCEL(QuitMacro) FileNew Type("Press Escape to Cancel") HardReturn HardReturn  FOR(x; 1; x &lt; 51; x + 1)     Type(x + NTOC(32))     SPEED(x/10)     IF(((x MOD 10) = 0) AND (x &lt; 50))         Slower </pre> |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

ENDIF
ENDFOR
QuitMacro

LABEL(Slower)
BEEP
HardReturn
Type("Slower ... ")
RETURN

LABEL(QuitMacro)
MESSAGEBOX(x; "SPEED EXAMPLE"; "Counting to 50, SPEED increased the delay between
counts."; IconInformation!)
QUIT

```

---

**See Also**

- ◆ WAIT

---

## STRLEN

---

**Purpose**

Determines the number of characters in a string.

---

**Syntax**

ReturnValue := STRLEN (*String*)

---

**Returns**

The number of characters in a string.

---

**Description**

The string can be a variable, constant, character string, or result of an expression.

```

vWord := "WordPerfect"
vNumber := STRLEN(vWord)

```

Result: vNumber equals 11

```

vNumber := STRLEN(45899)

```

Result: vNumber equals 5

```

vNumber := STRLEN("WordPerfect")

```

Result: vNumber equals 11

```

vNumber := STRLEN(9 + 9)

```

Result: vNumber equals 2

---

## Parameters

*String*

Character Expression: A variable, constant, character string, or result of an expression.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input for Y or N
// For Cancel condition: chose Cancel, click Close, press Alt+F4, or double-click system menu box
// Demonstrate STRLEN command

CANCEL(On!)
ONCANCEL CALL(CancelMessage)
PROCEDURE Message(vPrompt; vlcon)
 IF(vIcon = "!")
 BEEP
 MESSAGEBOX(y; "Message Box"; vPrompt; IconExclamation!)
 ELSE
 MESSAGEBOX(y; "Message Box"; vPrompt; IconInformation!)
 ENDIF
ENDPROC

vStr := ""
REPEAT
 GETSTRING(vStr; "Press one character: ""Y"" for Yes or ""N"" for no"; "GETSTRING - STRLEN
EXAMPLE"; 50)
 IF(vStr = "")
 vStr := "<Enter>"
 ENDIF

 vTest := TOUPPER(vStr)
 IF(STRLEN(vStr) > 1)
 CALL Message("You typed or pressed "" + vStr + """"; "!")
 ELSE
 IF((vTest = "Y") OR (vTest = "N"))
 CALL Message("Well done - You pressed " + vTest; "i")
 ELSE
 CALL Message("You pressed "" + vStr + """"; "!")
 ENDIF
 ENDIF
UNTIL(((vTest = "Y") OR (vTest = "N")) AND NOT(STRLEN(vStr) > 1))
QUIT

LABEL(CancelMessage)
vStr := "a Cancel control"
RETURN
```

---

## See Also

♦ CHARLEN ♦ NUMSTR ♦ STRNUM ♦ STRPOS ♦ STRUNIT ♦ SUBSTR

---

# STRNUM

---

**Purpose** Converts a character string of numbers to a numeric equivalent.

---

**Syntax** ReturnValue := STRNUM (*String*)

---

**Returns** A number.

```
vNum := STRNUM("123")
```

Result: vNum equals 123

```
vNum := STRNUM("123.5")
```

Result: vNum equals 123.5

```
vNum := STRNUM("9 + 9")
```

Result: vNum equals 9

---

## Parameters

*String*

Character expression: Contains a character string of numbers. STRNUM recognizes the decimal point defined by the sDecimal setting in the [intl] section of the Windows WIN.INI file. Alphabetic characters, operators, and punctuation marks, and everything that follows them are ignored.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract string of numbers and convert to numeric equivalent
// Demonstrate STRNUM command

vNumbers := ""
vPrompt := "Enter text that includes both letters and numbers:"
LABEL(Start)
GETSTRING(vStr; vPrompt; "EXTRACT NUMBERS")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vLen := STRLEN(vStr)
vPos := 1
REPEAT
 vTest := SUBSTR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest
 ENDIF
 vPos := vPos + 1
UNTIL(vPos = vLen + 1)
IF(STRLEN(vNumbers) > 0)
 vConvert := STRNUM(vNumbers)
 vConvert := vConvert * 5
```

```

 MESSAGEBOX(x; "STRNUM COMMAND"; vNumbers + " * 5 equals " + vConvert;
 IconInformation!)
 ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "You didn't enter any numbers"; IconExclamation!)
 ENDIF

```

## See Also

♦ NUMSTR ♦ STRLEN ♦ STRPOS ♦ STRUNIT ♦ SUBSTR

# STRPOS

**Purpose** Determines whether a character string is also a substring.

**Syntax** ReturnValue := STRPOS (*String*; *SubString*)

**Returns** The beginning position of a substring, or zero if a substring is not found.

```
vPos := STRPOS("WordPerfect"; "Perfect")
```

Result: vPos equals 5

```
vPos := STRPOS("WordPerfect"; "Scott")
```

Result: vPos equals 0

## Parameters

*String* Character Expression: A character string to evaluate.

*Substring* Character Expression: A substring to locate in *String*.

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract "Perfect" from WordPerfect and test if "Corporation" was typed
// Demonstrate STRPOS and SUBSTR commands

vNumbers := ""
LABEL(Start)
GETSTRING(vStr; "Enter the following text: ""WordPerfect Corporation""; "SUBSTR AND STRPOS
EXAMPLES")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vSub := SUBSTR(vStr; 5; 7)
IF(vSub = "Perfect")
 MESSAGEBOX(x; "SUBSTR EXAMPLE"; "WordPerfect is simply """" + vSub + """";
 IconInformation!)
 IF(STRPOS(vStr; "Corporation") = 0)

```

```

 BEEP

 MESSAGEBOX(x; "STRPOS EXAMPLE"; "You entered ""WordPerfect"" but not
 ""Corporation"""; IconExclamation!)
 ENDIF
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; ""WordPerfect"" wasn't the first word"; IconExclamation!)
 GO(Start)
ENDIF

```

---

**See Also**

• NUMSTR • STRLEN • STRNUM • STRUNIT • SUBSTR

---

## STRUNIT

---

**Purpose** Converts a string of numbers to a measurement.

---

**Syntax** ReturnValue := STRUNIT (*String*)

---

**Returns** A unit of measure.

---

**Description** The string may contain a number, an arithmetic expression that results in a number, or a character string of numbers. If a character string is used, alphabetic characters (except units of measure), operators, and punctuation marks are ignored. The default unit of measure is WordPerfect units. You can change the default with DEFAULTUNITS, or by specifying a unit of measure as part of the character string.

```

 DEFAULTUNITS(Centimeters)
 vUnit := STRUNIT(9)

```

Result: vUnit equals 9

```

 vUnit := STRUNIT(4+3)

```

Result: vUnit equals 7

```

 vUnit := STRUNIT("10")

```

Result: vUnit equals 10

```

 vUnit := STRUNIT("10abc")

```

Result: vUnit equals 10

```

 vUnit := STRUNIT("10i")

```

Result: vUnit equals 10

The units of measure are,

" inches  
i inches  
c centimeters  
m millimeters  
p points (72 per inch)  
w WP units (1200 per inch)

---

## Parameters

*String*

Character Expression: A number or character string of numbers.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract string of numbers and convert to measurement equivalent
// Demonstrate STRUNIT command

Retry := 4
LABEL(Start)
 vNumbers := ""
 GETSTRING(vStr; "Enter text that includes both letters and numbers."; "EXTRACT NUMBERS")
 IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
 ENDIF
 vLen := STRLEN(vStr)
 vPos := 1
 REPEAT
 vTest := SUBSTR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest
 ENDIF
 vPos := vPos + 1
 UNTIL(vPos = vLen + 1)
 IF(STRLEN(vNumbers) > 0)
 CALL(DefaultUnit)
 vConvert := STRUNIT(vNumbers)
 vConvert := vConvert * 5
 MESSAGEBOX(x; "STRUNIT COMMAND"; vNumbers + " * 5 equals " + vConvert;
 IconInformation! | RetryCancel!)
 IF(x = Retry)
 GO(Start)
 ENDIF
 ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "You didn't enter any numbers"; IconExclamation!)
 ENDIF
QUIT

LABEL(DefaultUnit)
 MENU(vChoice; Digit;; {"Inches"; "Centimeters"; "Millimeters"; "Points"; "WP Units"})
```

```

SWITCH(vChoice)
 CASEOF 1: DEFAULTUNITS(Inches!)
 CASEOF 2: DEFAULTUNITS(Centimeters!)
 CASEOF 3: DEFAULTUNITS(Millimeters!)
 CASEOF 4: DEFAULTUNITS(Points!)
 CASEOF 5: DEFAULTUNITS(WPUnits!)
ENDSWITCH
RETURN

```

## See Also

• NUMSTR • STRLEN • STRNUM • STRPOS • SUBSTR • UNITSTR

# SUBCHAR

## Purpose

Extracts a substring from a character string.

## Syntax

ReturnValue := SUBCHAR (*String*; *Beginning*; *NumberOfChars*)

## Returns

A character string.

```
vSub := SUBCHAR("WordPerfect"; 1; 4) // vSub = "Word"
```

## Description

Use CHARPOS to locate a substring.

## Parameters

*String*

Character Expression: A character string to evaluate.

*Beginning*

Numeric Expression: The starting position of a substring.

*NumberOfChars*

Numeric Expression: The number of characters to extract.

## Example

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract "Perfect" from WordPerfect
// Test if "Corporation" was entered
// Demonstrate SUBCHAR command

vNumbers := ""
vPrompt := "Enter the following text: ""WordPerfect Corporation""

LABEL(Start)
GETSTRING(vStr; vPrompt; "USING SUBCHAR")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vSub := SUBCHAR(vStr; 5; 7)
IF(vSub = "Perfect")

```



```

MESSAGEBOX(x; "SUBCHAR EXAMPLE"; "WordPerfect is simply "" + vSub + """;
IconInformation!)
IF(CHARPOS(vStr; "Corporation") = 0)
 BEEP
 MESSAGEBOX(x; "CHARPOS EXAMPLE"; "You entered ""WordPerfect"" but not
""Corporation"""; IconExclamation!)
ENDIF
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; ""WordPerfect"" wasn't the first word"; IconExclamation!)
 GO(Start)
ENDIF

```

---

## See Also

♦ CHARLEN ♦ CHARPOS ♦ SUBSTR

---

# SUBSTR

**Purpose** Extracts a substring from a character string.

**Syntax** ReturnValue := SUBSTR (*String*; *Beginning*; *NumberOfChars*)

**Returns** A character string.

```
vSub := SUBSTR("WordPerfect"; 1; 4) // vSub = "Word"
```

**Description** Use STRPOS to locate a substring.

## Parameters

*String* Character Expression: A character string to evaluate.

*Beginning* Numeric Expression: The starting position of a substring.

*NumberOfChars* Numeric Expression: The number of characters to extract.

**Example**

```

APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract "Perfect" from WordPerfect and test if "Corporation" was typed
// Demonstrate STRPOS and SUBSTR commands

vNumbers := ""
LABEL(Start)
GETSTRING(vStr; "Enter the following text: ""WordPerfect Corporation""; "SUBSTR AND STRPOS
EXAMPLES")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vSub := SUBSTR(vStr; 5; 7)

```

```

IF(vSub = "Perfect")
 MESSAGEBOX(x; "SUBSTR EXAMPLE"; "WordPerfect is simply "" + vSub + """;
 IconInformation!)
IF(STRPOS(vStr; "Corporation") = 0)
 BEEP
 MESSAGEBOX(x; "STRPOS EXAMPLE"; "You entered ""WordPerfect"" but not
 ""Corporation"""; IconExclamation!)
ENDIF
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; ""WordPerfect"" wasn't the first word"; IconExclamation!)
GO(Start)
ENDIF

```

---

**See Also**

♦ NUMSTR ♦ STRLEN ♦ STRNUM ♦ STRPOS ♦ STRUNIT ♦ SUBCHAR

---

## SWITCH-ENDSWITCH

---

**Purpose**

A conditional statement that tests for matching expressions. If a match is found, a statement (or statement block) is executed (see *Conditional Statements in Macro Control Statements*).

---

**Syntax**

The general form of a SWITCH statement is,

```

SWITCH (Test)
 CASEOF Selector1:
 ...statement block...
 CONTINUE
 CASEOF Selector2; Selector3:
 ...statement block...
 DEFAULT:
 ...statement block...
ENDSWITCH

```

---

**Description**

If *Test* matches *Selector*, the statement block that follows *Selector* is executed and no other evaluation is made. *Test* and *Selector* are case sensitive and must match exactly.

If CONTINUE follows an executed statement block, the next statement block is automatically executed. CONTINUE is optional.

The DEFAULT statement block is executed if no *Selector* matches *Test*. DEFAULT is optional. If no match is found and DEFAULT is not used, the macro continues to the first statement after ENDSWITCH.

If BREAK occurs in a state block, the macro continues to the first statement after ENDSWITCH.

ENDSWITCH closes a SWITCH statement.

---

## Parameters

*Test*

Any: The control expression. Variables are assigned values by commands such as GETSTRING, GETNUMBER, or MENU.

*Selector*

Any: An expression (variable, constant, character) with a value that is usually assigned before the macro is compiled. It is possible to assign the value at run-time. *Selector* always follows a CASEOF statement.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Display message box on ASSERT condition
// Demonstrate SWITCH command

ONCANCEL(Condition)
ONERROR(Condition)
ONNOTFOUND(Condition)

LABEL(Start)
 MENU(vPick; Digit!;; {"Assert Cancel"; "Assert Error"; "Assert Not Found"; "Quit"})
 SWITCH(vPick)
 CASEOF 1: ASSERT(CancelCondition!)
 CASEOF 2: ASSERT(ErrorCondition!)
 CASEOF 3: ASSERT(NotFoundCondition!)
 CASEOF 4: QUIT
 DEFAULT: GO(Condition)
 ENDSWITCH

LABEL(Condition)
 SWITCH(ErrorNumber)
 CASEOF 1: vMsg := "1 (Cancel condition)"
 CASEOF 2: vMsg := "2 (Error condition)"
 CASEOF 7: vMsg := "3 (Not Found condition)"
 DEFAULT: vMsg := "?"
 ENDSWITCH
 MESSAGEBOX(vStatus; "ASSERT COMMAND"; "You selected " + vMsg; IconInformation!)
 GO(Start)
```

---

## See Also

♦ BREAK ♦ CASE ♦ CASE CALL ♦ IF

---

# TOLOWER

---

**Purpose** Converts uppercase letters to lowercase.

---

**Syntax** ReturnValue := TOLOWER (*String*)

---

**Returns** The lowercase equivalent of uppercase letters.

---

**Description** The value returned by TOLOWER can be assigned to a variable or used directly by a statement such as IF.

```
vAnswer := "YES"
vAnswer := TOLOWER(vAnswer) // vAnswer = lowercase yes
```

**Shorthand notation example**

```
IF(TOLOWER(vAnswer) = "yes") // Beep if vAnswer = lowercase yes
BEEP
ELSE
QUIT
ENDIF
```

---

**Parameters**

*String* Character Expression: A string to convert.

---

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate TOLOWER and VARERRCHK command
```

```
HdReturn := NTOC(0F90Ah)
VARERRCHK(Off)
```

```
FOR(x; 65; x < 91; x + 1)
 vLongStr := vLongStr + TOLOWER(NTOC(x)) + NTOC(32)
ENDFOR
```

```
MESSAGEBOX(x; "VARERRCHK and TOLOWER"; "With VARERRCHK(On!), you would receive a
run-time error message." + HdReturn + HdReturn + "vLongStr: " + vLongStr + HdReturn;
IconInformation!)
```

---

**See Also** ♦ TOUPPER

---

# TOUPPER

---

|                    |                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | Converts lowercase letters to uppercase.                                                                |
| <b>Syntax</b>      | ReturnValue := TOUPPER ( <i>String</i> )                                                                |
| <b>Returns</b>     | The uppercase equivalent of lowercase letters.                                                          |
| <b>Description</b> | The value returned by TOUPPER can be assigned to a variable or used directly by a statement such as IF. |

---

```
vAnswer := "yes"
vAnswer := TOUPPER(vAnswer) // vAnswer = uppercase YES
```

### Shorthand notation example

```
IF(TOUPPER(vAnswer) = "YES") // Beep if vAnswer = uppercase YES
BEEP
ELSE
QUIT
ENDIF
```

---

## Parameters

*String* Character Expression: A string to convert.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Test user input for Y or N
// For Cancel condition: chose Cancel, click Close, press Alt+F4, or double-click system menu box
// Demonstrate TOUPPER command

CANCEL(On!)
ONCANCEL.CALL(CancelMessage)
PROCEDURE Message(vPrompt; vIcon)
 IF(vIcon = "!")
 BEEP
 MESSAGEBOX(y; "Message Box"; vPrompt; IconExclamation!)
 ELSE
 MESSAGEBOX(y; "Message Box"; vPrompt; IconInformation!)
 ENDIF
ENDPROC

vStr := ""
REPEAT
 GETSTRING(vStr; "Press one character: ""Y"" for Yes or ""N"" for no"; "GETSTRING EXAMPLE";
 50)
 IF(vStr = "")
 vStr := "<Enter>"
 ENDIF
```

```

vTest := TOUPPER(vStr)
IF(CHARLEN(vStr) > 1)
 CALL Message("You typed or pressed "" + vStr + """"; "!")
ELSE
 IF((vTest = "Y") OR (vTest = "N"))
 CALL Message("Well done - You pressed " + vTest; "!")
 ELSE
 CALL Message("You pressed "" + vStr + """"; "!")
 ENDIF
ENDIF
UNTIL(((vTest = "Y") OR (vTest = "N")) AND NOT(CHARLEN(vStr) > 1))
QUIT

LABEL(CancelMessage)
vStr := "a Cancel control"
RETURN

```

---

**See Also**

♦ TOLOWER

---

## UNITSTR

---

**Purpose**

Converts a measurement to a character string.

---

**Syntax**

ReturnValue := UNITSTR (*Value; Units*)

---

**Returns**

A measurement expressed as a string.

---

**Description**

If a unit of measure is not specified, the default is WordPerfect units. To change the default use DEFAULTUNITS. The following example converts WP units to centimeters and returns the result as a string:

```
vUnits := UNITSTR(1200; Centimeters!)
```

Result: vUnits equals "2.540000C"

The next example converts centimeters to millimeters and returns the result as a string:

```
DEFAULTUNITS(Centimeters!)
vUnits := UNITSTR(1.0; Millimeters!)
```

Result: vUnits equals "10.000000M"

The next example converts inches to centimeters and returns the result as a string:

```
vUnits := UNITSTR(2.0i; Centimeters!)
```

Result: vUnits equals "5.080000C"

See STRUNIT to convert a string of numbers to a measurement.

---

## Parameters

|              |                                                                                                                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Value</i> | Measurement Expression: A real number measurement. Unless you specify a unit of measure, the default is WP Units. To change the default use DEFAULTUNITS.                                     |
| <i>Units</i> | Enumeration: The unit of measure that <i>Value</i> converts to.<br>Inches!<br>Centimeters!<br>Millimeters!<br>Points! (72 per inch)<br>WPUnits! (1200 per inch)<br>WP1200ths! (1200 per inch) |

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate UNITSTR and DIALOG commands

vNbr := 1
vStatus := 4
DIALOGDEFINE(1000; 50; 50; 100; 120; Percent! | NoTitle! | NoFrame!; "")
 DIALOGADDDTEXT(1000; -1; 10; 15; 75; 15; Left!; "Convert " + vNbr + " inch to:")
 DIALOGADDCHECKBOX(1000; 101; 15; 30; 100; 13; "Centimeters"; vCB1)
 DIALOGADDCHECKBOX(1000; 102; 15; 45; 100; 13; "Millimeters"; vCB2)
 DIALOGADDCHECKBOX(1000; 103; 15; 60; 100; 13; "Points"; vCB3)
 DIALOGADDCHECKBOX(1000; 104; 15; 75; 100; 13; "WP Units"; vCB4)
 DIALOGADDDTEXT(1000; -2; 10; 100; 100; 15; Left!; "(Press Alt+F4 to cancel)")
DIALOGDISPLAY(1000; -1; Msg)

WHILE(vStatus = 4)
ENDWHILE
DIALOGDESTROY(1000)

LABEL(Msg)
 IF(Msg[5] = 274)
 BEEP
 vStatus := 2
 ENDIF
 DEFAULTUNITS(Inches!)
 SWITCH(Msg[3])
 CASEOF 101: vResult := UNITSTR(vNbr; Centimeters!)
 Message
 CASEOF 102: vResult := UNITSTR(vNbr; Millimeters!)
 Message
 CASEOF 103: vResult := UNITSTR(vNbr; Points!)
 Message
 CASEOF 104: vResult := UNITSTR(vNbr; WPUnits!)
 Message
 ENDSWITCH
 RETURN

LABEL(Message)
 DIALOGUNDISPLAY(1000; "CancelBttn")
```

```
MESSAGEBOX(vStatus; "UNITSTR COMMAND"; vNmbr + " inch is converted to " + vResult;
IconInformation! | RetryCancel!)
IF(vStatus = 4)
 DIALOGDISPLAY(1000; -1; Msg)
ELSE
 BEEP
ENDIF
RETURN
```

---

**See Also**

♦ GETUNITS ♦ NUMSTR ♦ STRLEN ♦ STRNUM ♦ STRPOS ♦ SUBSTR  
♦ STRUNIT

---

## USE

---

**Purpose**

Enables macros to share functions and procedures.

---

**Syntax**

USE (*MacroFile*)

---

**Description**

USE is a non-executable statement that can occur anywhere in a macro. USE identifies a macro file with functions and/or procedures that can be called from another macro. A macro with a calling statement to a function or procedure in another macro file must include a USE statement that identifies the file. A macro that includes only function and/or procedure statements must be compiled like any macro file (see Macro File Library).

---

**Parameters***MacroFile*

Character Expression: The path and name of a macro file. You cannot use a variable.

---

**Example**

```
// Macro File Library (MACROLIB.WCM)
// Macro Facility displays a Warning Error when you compile this macro. For each warning, choose
Continue Compilation.
// There are two macros in this example.
// Demonstrate USE command

APPLICATION(A1; "WordPerfect"; Default; "US")
FUNCTION GetMsgBoxResult(x)
 VARERRCHK(Off!)
 BEEP
 SWITCH(x)
```



```

CASEOF 1: MESSAGEBOX(vChoice; "Macro File Library"; "Choose a button to dismiss the
message box"; IconStop! | AbortRetryIgnore!)
CASEOF 2: MESSAGEBOX(vChoice; "Macro File Library"; "Choose a button to dismiss the
message box"; IconStop! | Ok!)
CASEOF 3: MESSAGEBOX(vChoice; "Macro File Library"; "Choose a button to dismiss the
message box"; IconStop! | OkCancel!)
CASEOF 4: MESSAGEBOX(vChoice; "Macro File Library"; "Choose a button to dismiss the
message box"; IconStop! | RetryCancel!)
CASEOF 5: MESSAGEBOX(vChoice; "Macro File Library"; "Choose a button to dismiss the
message box"; IconStop! | YesNo!)
CASEOF 6: MESSAGEBOX(vChoice; "Macro File Library"; "Choose a button to dismiss the
message box"; IconStop! | YesNoCancel!)
ENDSWITCH
RETURN(vChoice)
ENDFUNC

PROCEDURE DisplayMsgBoxResult(x)
MESSAGEBOX(x; "Macro File Library"; "You chose the "" + x + "" button"; IconInformation!)
ENDPROC

APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate USE command

VARERRCHK(Off!)
USE("C:\WPWIN60\MACROS\MACROLIB.WCM")
REPEAT
MENU(vResult; Digit;; {"AbortRetryIgnore!"; "OK!"; "OKCancel!"; "RetryCancel!"; "YesNo!";
"YesNoCancel!"; "Quit"})
SWITCH(GetMsgBoxResult(vResult))
CASEOF 1: vStatus = "OK"
CASEOF 2: vStatus = "Cancel"
CASEOF 3: vStatus = "Abort"
CASEOF 4: vStatus = "Retry"
CASEOF 5: vStatus = "Ignore"
CASEOF 6: vStatus = "Yes"
CASEOF 7: vStatus = "No"
ENDSWITCH
IF(NOT(vResult = 7))
DisplayMsgBoxResult(vStatus)
ENDIF
UNTIL(vResult = 7)

```

---

## See Also

♦ FUNCTION ♦ PROCEDURE

---

# VARERRCHK

---

**Purpose** Determines how a macro responds to uninitialized variables (variables not assigned a value).

**Syntax** VARERRCHK (*State*)

**Parameters**

*State* Enumeration: Specifies the state of variable checking. The default is VARERRCHK(On!). The enumerations are,

- Off! Ignore uninitialized variables by temporarily assigning a value of 0.
- On! Display a run-time Error message when a macro attempts to use an uninitialized variable, and end the macro.

**Example**

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Demonstrate TOLOWER and VARERRCHK command

HdReturn := NTOC(0F90Ah)
VARERRCHK(Off!)

FOR(x; 65; x < 91; x + 1)
 vLongStr := vLongStr + TOLOWER(NTOC(x)) + NTOC(32)
ENDFOR

MESSAGEBOX(x; "VARERRCHK and TOLOWER"; "With VARERRCHK(On!), you would receive a
run-time error message." + HdReturn + HdReturn + "vLongStr: " + vLongStr + HdReturn;
IconInformation!)
```

**See Also** ♦ GLOBAL ♦ LOCAL ♦ PERSIST ♦ PERSISTALL

---

---

# WAIT

---

**Purpose** Pauses macro execution.

**Syntax** WAIT (*TenthsOfSeconds*)

**Description** Time is measured in tenths of a second. The maximum pause is one minute, or WAIT(600).

---

## Parameters

*TenthsOfSeconds*

Numeric Expression: A number from zero to 600. Divide the number by 10 to calculate the number of seconds.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Play .wav file
// Assume Windows sound system and that appropriate driver is installed
// Demonstrate MMPLAY and WAIT commands

No := 7
REPEAT
 MENU(vWav; Digit;; {"Siren"; "Crickets"; "Snoring"; "Thunder"; "Dog bark"; "Elephant"; "Quit"})
 SWITCH(vWav)
 CASEOF 1: PlaySound("siren.wav"; 115)
 CASEOF 2: PlaySound("crickets.wav"; 50)
 CASEOF 3: PlaySound("snoring.wav"; 30)
 CASEOF 4: PlaySound("thunder.wav"; 30)
 CASEOF 5: PlaySound("dogbark.wav"; 15)
 CASEOF 6: PlaySound("elephant.wav"; 15)
 CASEOF 7: QUIT
 ENDSWITCH
 MESSAGEBOX(vStatus; "MMPLAY"; "Would you like to hear another sound?"; IconQuestion! |
 YesNo!)
UNTIL(vStatus = No)

PROCEDURE PlaySound(vSound; vWait)
 MMPLAY("c:\sndsys\sounds\" + vSound)
 WAIT(vWait)
ENDPROC
```

---

## See Also

- ♦ SPEED

---

# WHILE-ENDWHILE

---

**Purpose** A loop statement that executes while the expression at the top of the loop is true. See Loop Statements.

---

**Syntax** The general form of a WHILE statement is,

```
WHILE (Test)
...statement block...
ENDWHILE
```

---

**Description** The loop does not execute the first time unless *Test* is true. When *Test* is false, the first statement after ENDWHILE is executed.

---

---

## Parameters

*Test*

Relational Expression: Evaluates to true or false.

---

## Example

```
APPLICATION(A1; "WordPerfect"; Default; "US")
// Extract numbers from a character string
// Demonstrate WHILE command

vNumbers := ""
vPrompt := "Enter text that includes both letters and numbers:"

LABEL(Start)
GETSTRING(vStr; vPrompt; "EXTRACT NUMBERS")
IF(vStr = "")
 BEEP
 MESSAGEBOX(x; "ERROR"; "You pressed return without entering any text"; IconStop!)
 GO(Start)
ENDIF
vLen := CHARLEN(vStr)
vPos := 1
WHILE(NOT(vPos = vLen + 1))
 vTest := SUBCHAR(vStr; vPos; 1)
 IF((CTON(vTest) > 47) AND (CTON(vTest) < 58))
 vNumbers := vNumbers + vTest
 ENDIF
 vPos := vPos + 1
ENDWHILE
IF(CHARLEN(vNumbers) > 0)
 MESSAGEBOX(x; "NUMBER STRING"; "You entered the following numbers: " + vNumbers;
 IconInformation!)
ELSE
 BEEP
 MESSAGEBOX(x; "ERROR"; "You didn't enter any numbers"; IconExclamation!)
ENDIF
```

---

## See Also

♦ FOR ♦ FOREACH ♦ FORNEXT ♦ REPEAT

# Product Commands Reference

.....



---

# AbbreviationCreate

---

|                         |                                                                                                                                                                                                                 |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>          | AbbreviationCreate creates an abbreviation in a template.                                                                                                                                                       |
| <b>Syntax</b>           | AbbreviationCreate (AbbreviationName: <i>Character Expression</i> ; Template: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Text: <i>Character Expression</i> )                                         |
| <b>Parameters</b>       |                                                                                                                                                                                                                 |
| <i>AbbreviationName</i> | Character Expression: Specifies the name of an abbreviation.                                                                                                                                                    |
| <i>Template</i>         | Numeric Equivalent or Enumerated Type: Specifies the template containing the abbreviation.<br><ul style="list-style-type: none"><li>0 CurrentDoc!</li><li>1 PersonalLibrary!</li><li>2 SharedLibrary!</li></ul> |
| <i>Text</i>             | Character Expression: Specifies the text assigned to an abbreviation name.                                                                                                                                      |
| <b>Route</b>            | Select text, Insert, Abbreviations, Create, enter an abbreviation name.                                                                                                                                         |
| <b>Example</b>          | To set the abbreviation for "WordPerfect" to "WP" in the current template, the command is:<br><pre>AbbreviationCreate (AbbreviationName: "WP"; Template: CurrentDoc!; Text: "WordPerfect")</pre>                |
| <b>See Also</b>         | • AbbreviationDelete • AbbreviationExpand                                                                                                                                                                       |

---

# AbbreviationDelete

---

|                         |                                                                                                                                                                                                                 |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>          | AbbreviationDelete deletes an abbreviation from a template.                                                                                                                                                     |
| <b>Syntax</b>           | AbbreviationDelete (AbbreviationName: <i>Character Expression</i> ; Template: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                             |
| <b>Parameters</b>       |                                                                                                                                                                                                                 |
| <i>AbbreviationName</i> | Character Expression: Specifies an abbreviation to delete.                                                                                                                                                      |
| <i>Template</i>         | Numeric Equivalent or Enumerated Type: Specifies the template containing the abbreviation.<br><ul style="list-style-type: none"><li>0 CurrentDoc!</li><li>1 PersonalLibrary!</li><li>2 SharedLibrary!</li></ul> |

|                 |                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Insert, Abbreviations, select an abbreviation, Delete.                                                                                           |
| <b>Example</b>  | To delete the abbreviation DBW from the current template, the command is:<br>AbbreviationDelete (AbbreviationName: "DBW"; Template: CurrentDoc!) |
| <b>See Also</b> | ♦ AbbreviationCreate ♦ AbbreviationExpand                                                                                                        |

---

## AbbreviationDlg

|                |                                                       |
|----------------|-------------------------------------------------------|
| <b>Purpose</b> | AbbreviationDlg displays the Abbreviation dialog box. |
| <b>Syntax</b>  | AbbreviationDlg ()                                    |
| <b>Route</b>   | Insert, Abbreviations.                                |

---

## AbbreviationExpand

|                |                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | AbbreviationExpand expands an abbreviation.                                                                                  |
| <b>Syntax</b>  | AbbreviationExpand (AbbreviationName: <i>Character Expression</i> ; Template: <i>Numeric Equivalent or Enumerated Type</i> ) |

### Parameters

|                         |                                                                                                                                                          |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>AbbreviationName</i> | Character Expression: Specifies the name of an abbreviation to expand.                                                                                   |
| <i>Template</i>         | Numeric Equivalent or Enumerated Type: Specifies the template that contains the abbreviation.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary! |

|                |                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Insert, Abbreviations, select an abbreviation, Expand.                                                                                                |
| <b>Example</b> | To expand "bldg" to "building" in the Personal Library, the command is:<br>AbbreviationExpand: (AbbreviationName: "bldg"; Template: PersonalLibrary!) |



---

**See Also**

♦ AbbreviationCreate ♦ AbbreviationDelete

---

## AboutDlg

---

**Purpose**

AboutDlg displays the About WordPerfect dialog box.

**Syntax**

AboutDlg ()

**Route**

Help, About WordPerfect.

---

## Advance

---

**Purpose**

Advance positions the insertion point a specified distance from the top or left of the page or at a specified distance from the insertion point.

**Syntax**

Advance (Where: *Numeric Equivalent* or *Enumerated Type*, Amount: *Measurement Expression*)

**Parameters***Where*

Numeric Equivalent or Enumerated Type: Specifies the text advance type.

- 0 AdvanceLeft!
- 1 AdvanceRight!
- 2 AdvanceFromLeftEdge!
- 3 AdvanceUp!
- 4 AdvanceDown!
- 5 AdvanceFromTop!

*Amount*

Measurement Expression: Specifies an advance distance.

**Route**

Layout, Typesetting, Advance, set options.

**Example**

To position text 3.5" to the right of the insertion point, the command is:

Advance (Where: AdvanceRight!; Amount: 3.5")

**See Also**

♦ AdvanceDlg

---

---

## AdvanceDlg

|                |                                             |
|----------------|---------------------------------------------|
| <b>Purpose</b> | AdvanceDlg displays the Advance dialog box. |
| <b>Syntax</b>  | AdvanceDlg ()                               |
| <b>Route</b>   | Layout, Typesetting, Advance.               |

---

## Append

|                |                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | Append adds a copy of selected text or graphics to the end of the clipboard contents.<br>IMPORTANT: Macro execution terminates if you use this command when the clipboard is empty. |
| <b>Syntax</b>  | Append ()                                                                                                                                                                           |
| <b>Route</b>   | Select text or graphics, Edit, Append.                                                                                                                                              |

---

## AppMaximize

|                 |                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | AppMaximize maximizes an application window.<br>IMPORTANT: If already maximized, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | AppMaximize ()                                                                                                                                                                     |
| <b>Route</b>    | AppMaximize is the same as choosing Maximize from the WordPerfect Control menu or clicking the WordPerfect Maximize button.                                                        |
| <b>See Also</b> | ♦ AppMinimize ♦ AppRestore ♦ DocMaximize ♦ DocMinimize                                                                                                                             |

---

## AppMinimize

|                |                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | AppMinimize minimizes an application window to an icon.<br>IMPORTANT: If already minimized, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                                        |
|-----------------|--------------------------------------------------------|
| <b>Syntax</b>   | AppMinimize ()                                         |
| <b>Route</b>    | Choose Minimize from the WordPerfect control menu.     |
| <b>See Also</b> | ♦ AppMaximize ♦ AppRestore ♦ DocMaximize ♦ DocMinimize |

---

## AppMove

---

|                 |                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | AppMove repositions a window.<br><br>IMPORTANT: If WordPerfect is maximized, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | AppMove ()                                                                                                                                                  |
| <b>Route</b>    | Choose Move from the WordPerfect control menu and reposition the window as desired.                                                                         |
| <b>See Also</b> | ♦ DocMove                                                                                                                                                   |

---

## AppRestore

---

|                 |                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | AppRestore restores a window to its previous size.<br><br>IMPORTANT: If WordPerfect has already been restored, this command terminates the macro or redirects macro execution to the LABEL command specified by ONERROR. |
| <b>Syntax</b>   | AppRestore ()                                                                                                                                                                                                            |
| <b>Route</b>    | Choose Restore from the WordPerfect control menu.                                                                                                                                                                        |
| <b>See Also</b> | ♦ AppMaximize ♦ AppMinimize ♦ DocRestore                                                                                                                                                                                 |

---

---

## AppSize

|                 |                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | AppSize resizes a window.<br>IMPORTANT: If already maximized, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | AppSize ()                                                                                                                                   |
| <b>Route</b>    | Choose Size from the WordPerfect control menu.                                                                                               |
| <b>See Also</b> | • DocSize                                                                                                                                    |

---

## AppTaskList

|                |                                                     |
|----------------|-----------------------------------------------------|
| <b>Purpose</b> | AppTaskList displays the Windows Task List.         |
| <b>Syntax</b>  | AppTaskList ()                                      |
| <b>Route</b>   | Choose Switch To from the WordPerfect control menu. |

---

## AssociateDlg

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| <b>Purpose</b> | AssociateDlg displays the Associate dialog box.                       |
| <b>Syntax</b>  | AssociateDlg ()                                                       |
| <b>Route</b>   | File, Template, Options, Edit Template or Create Template, Associate. |

---

## AttributeAppearanceOff

|                |                                                                                            |
|----------------|--------------------------------------------------------------------------------------------|
| <b>Purpose</b> | AttributeAppearanceOff deselects specified attribute(s) where this command is encountered. |
| <b>Syntax</b>  | AttributeAppearanceOff ({Attrib: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> })    |

---

## Parameters

### *Attrib*

Numeric Equivalent or Enumerated Type: Specifies attributes to deselect. `AttributeAppearanceOff` requires at least one attribute parameter. Enclose multiple attributes in braces ({}), separated by semicolons.

- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!
- 16 Every!

---

## Route

Layout, Font, deselect attributes.

---

## Example

To turn off Shadow, Underline, and Small Caps, the command is:

```
AttributeAppearanceOff ({Attrib: Shadow!; Attrib: Underline!; Attrib: Small Caps!})
```

---

## See Also

♦ `AttributeAppearanceOn` ♦ `AttributeNormal`

---

# AttributeAppearanceOn

---

## Purpose

`AttributeAppearanceOn` selects specified attributes where this command is encountered.

---

## Syntax

`AttributeAppearanceOn` ({Attrib: *Numeric Equivalent or Enumerated Type*})

---

## Parameters

### *Attrib*

Numeric Equivalent or Enumerated Type: Specifies attributes to select. `AttributeAppearanceOn` requires at least one attribute parameter. Enclose multiple attributes in braces ({}), separated by commas.

- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!
- 16 Every!

|                 |                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Font, select attributes.                                                                                                                        |
| <b>Example</b>  | To turn on Shadow, Bold, and Small Caps, the command is:<br><code>AttributeAppearanceOn ({Attrib: Shadow!; Attrib: Bold!; Attrib: Small Caps!})</code>  |
| <b>See Also</b> | ♦ <code>AttributeAppearanceOff</code> ♦ <code>AttributeNormal</code> ♦ <code>AttributeRelativeSize</code><br>♦ <code>AttributeRelativeSizeToggle</code> |

---

## AttributeAppearanceToggle

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | <code>AttributeAppearanceToggle</code> toggles attributes, usually selected of text.                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Syntax</b>     | <code>AttributeAppearanceToggle</code> ( <i>{Attrib: Numeric Equivalent or Enumerated Type}</i> )                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Parameters</b> | <p><i>Attrib</i></p> <p>Numeric Equivalent or Enumerated Type: Specifies attributes to toggle. <code>AttributeAppearanceToggle</code> requires at least one attribute parameter. Enclose multiple attributes in braces ({} ) separated by semicolons.</p> <ul style="list-style-type: none"> <li>7 Outline!</li> <li>8 Italics!</li> <li>9 Shadow!</li> <li>10 Redline!</li> <li>11 DoubleUnderline!</li> <li>12 Bold!</li> <li>13 Strikeout!</li> <li>14 Underline!</li> <li>15 SmallCaps!</li> <li>16 Every!</li> </ul> |

|                 |                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Select an attribute appearance button on a Button Bar or Power Bar.                                                                                                                          |
| <b>Example</b>  | To toggle Italics and Redline, the command is:<br><code>AttributeAppearanceToggle ({Attrib: Italics!; Attrib: Redline!})</code>                                                              |
| <b>See Also</b> | ♦ <code>AttributeAppearanceOff</code> ♦ <code>AttributeAppearanceOn</code> ♦ <code>AttributeNormal</code><br>♦ <code>AttributeRelativeSize</code> ♦ <code>AttributeRelativeSizeToggle</code> |

---

## AttributeNormal

---

|                 |                                                                                                                                                                                                                                   |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | AttributeNormal deselects current font attributes. This command does not reset font color.<br><br>IMPORTANT: To record this command, you must first add an Attribute Normal button to the Button Bar or Power Bar in Preferences. |
| <b>Syntax</b>   | AttributeNormal ()                                                                                                                                                                                                                |
| <b>Route</b>    | Attribute Normal on a Button Bar or Power Bar.                                                                                                                                                                                    |
| <b>See Also</b> | ♦ AttributeAppearanceOff ♦ AttributeAppearanceOn ♦ AttributeRelativeSize<br>♦ AttributeRelativeSizeToggle                                                                                                                         |

---

## AttributePosition

---

|                   |                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | AttributePosition adjusts the vertical position of subsequent or selected text.                                                                                   |
| <b>Syntax</b>     | AttributePosition (Position: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                |
| <b>Parameters</b> |                                                                                                                                                                   |
| <i>Position</i>   | Numeric Equivalent or Enumerated Type: Specifies the vertical position of subsequent or selected text.<br><br>5 Superscript!<br>6 Subscript!<br>7 NormalPosition! |
| <b>Route</b>      | Layout, Font, choose Superscript, Normal, or Subscript from the Position pop-up list.                                                                             |
| <b>Example</b>    | To set the position of subsequent or selected text as superscript, the command is:<br><br>AttributePosition (Position: Superscript!)                              |
| <b>See Also</b>   | ♦ AttributeNormal ♦ AttributePositionToggle                                                                                                                       |

---

# AttributePositionToggle

**Purpose** AttributePositionToggle toggles the specified position.

**Syntax** AttributePositionToggle (Position: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*Position* Numeric Equivalent or Enumerated Type: Specifies a position to toggle.

- 5 Superscript!
- 6 Subscript!
- 7 NormalPosition!

**Route**

Choose an attribute position button on a Button Bar or Power Bar.

**Example**

To toggle a subscript position, the command is:

AttributePositionToggle (Position: Subscript!)

**See Also**

♦ AttributeNormal ♦ AttributePosition

---

# AttributeRelativeSize

**Purpose** AttributeRelativeSize specifies a relative font size for subsequent or select text.

**Syntax** AttributeRelativeSize (Size: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*Size* Numeric Equivalent or Enumerated Type: Specifies the relative size of subsequent or selected text.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 NormalSize!

**Route**

Layout, Font, Relative Size, select a size.



---

**Example** To select Large, the command is:  
`AttributeRelativeSize (Size: Large!)`

---

**See Also** ♦ `AttributeRelativeSizeToggle`

---

---

## AttributeRelativeSizeToggle

---

**Purpose** `AttributeRelativeSizeToggle` toggles a relative size.

---

**Syntax** `AttributeRelativeSizeToggle` (Size: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Size* Numeric Equivalent or Enumerated Type: Specifies a relative size to toggle.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 NormalSize!

---

**Route** A relative size button on a Button Bar or Power Bar.

---

**Example** To toggle Fine, the command is:  
`AttributeRelativeSizeToggle (Size: Fine!)`

---

**See Also** ♦ `AttributeRelativeSize`

---

---

## AttributeSizeRatios

---

**Purpose** `AttributeSizeRatios` changes the relative size ratio of a font selected by `AttributeRelativeSize`.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** `AttributeSizeRatios` (Attrib: *Numeric Equivalent* or *Enumerated Type*, Percentage: *Numeric Expression*)

---

---

## Parameters

*Attrib* Numeric Equivalent or Enumerated Type: Specifies a size to edit.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 SuperSubScript!

*Percentage* Numeric Expression: Specifies the relative size of a font selected by `AttributeRelativeSize`.

---

## Example

To specify a size ratio for Large that is 150 percent of normal, the command is:

```
AttributeSizeRatios (Attrib: Large!; Percentage: 150)
```

---

## See Also

♦ `AttributeRelativeSize` ♦ `AttributeRelativeSizeToggle`

---

# BackTab

---

## Purpose

`BackTab` inserts a Hard Back Tab [Hd Back Tab] code at the insertion point.

---

## Syntax

```
BackTab ()
```

---

## Route

Layout, Paragraph, Back Tab.

---

# Backup

---

## Purpose

`Backup` specifies whether Timed Document Backup is on.

---

## Syntax

```
Backup (State: Numeric Equivalent or Enumerated Type)
```

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether Backup is on.

- 0 Off!
- 1 On!

---

**Route** File, Preferences, File, Documents/Backup, Timed Document Backup every \_\_ minutes.

---

**Example** To turn off Backup, the command is:  
Backup (State: Off!)

---

**See Also** ♦ BackupMinutes ♦ BackupOriginalDoc

---

---

## BackupMinutes

---

**Purpose** BackupMinutes specifies the number of minutes between each timed backup.

---

**Syntax** BackupMinutes (Minutes: *Numeric Expression*)

---

**Parameters**

*Minutes* Numeric Expression: Specifies the number of minutes between backups.

---

**Route** File, Preferences, File, select Documents/Backup, enter a number in the Timed Document Backup every \_\_ minutes counter box.

---

**Example** To set a timed backup interval of 15 minutes, the command is:  
BackupMinutes (Minutes: 15)

---

**See Also** ♦ Backup ♦ BackupOriginalDoc

---

---

## BackupOriginalDoc

---

**Purpose** BackupOriginalDoc specifies whether Original Document Backup is on.

---

**Syntax** BackupOriginalDoc (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type: Specifies whether Original Document Backup is on.

0 Off!  
1 On!

---

|                 |                                                                                        |
|-----------------|----------------------------------------------------------------------------------------|
| <b>Route</b>    | File, Preferences, File icon, Original Document Backup.                                |
| <b>Example</b>  | To turn on Original Document Backup, the command is:<br>BackupOriginalDoc (State: On!) |
| <b>See Also</b> | ♦ Backup ♦ BackupMinutes ♦ PrefSave                                                    |

## BarcodeDlg

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| <b>Purpose</b>  | BarcodeDlg displays the POSTNET Barcode dialog box. |
| <b>Syntax</b>   | BarcodeDlg ()                                       |
| <b>Route</b>    | Insert, Other, Bar Code.                            |
| <b>See Also</b> | ♦ BarCodePOSTNET                                    |

## BarcodePOSTNET

|                   |                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BarcodePOSTNET specifies a ZIP Code or a merge field containing a ZIP Code.                                                                                                                                                  |
| <b>Syntax</b>     | BarcodePOSTNET (ZIP: <i>Numeric Expression</i> or <i>Character Expression</i> ; State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                                                              |
| <i>ZIP</i>        | Numeric Expression or Character Expression: Specifies a five, nine, or eleven-digit ZIP Code. If the ZIP Code is contained in a merge field, this parameter can be a character expression specifying the merge field to use. |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether a merge field is used to provide the ZIP Code. This parameter is necessary only when a merge field has been specified in the ZIP parameter.<br>1 MergeFieldAllowed! |
| <b>Route</b>      | Insert, Other, Bar Code, enter bar code contents.                                                                                                                                                                            |
| <b>Example</b>    | To enter a POSTNET barcode in a document, the command is:<br>BarcodePOSTNET (ZIP: "84057")                                                                                                                                   |

---

**See Also**

- ♦ BarCodeDlg

---

## BaselinePlacement

---

**Purpose** BaselinePlacement specifies whether Baseline Placement for Typesetting is on.

---

**Syntax** BaselinePlacement (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type: Specifies whether Baseline Placement for Typesetting is on.

- 0 Off!
- 1 On!

---

**Route** Layout, Typesetting, Word/Letterspacing, Baseline Placement for Typesetting.

---

**Example** To turn on Baseline Placement for Typesetting, the command is:

```
BaselinePlacement (State: On!)
```

---

**See Also**

- ♦ ?BaselineMode

---

## Binding

---

**Purpose** Binding specifies the binding offset and the edge of the paper to which that offset should be applied. The offset will be added to the margin of the specified edge.

---

**Syntax** Binding (Edge: *Numeric Equivalent* or *Enumerated Type*, Offset: *Measurement Expression*)

---

**Parameters**

*Edge* Numeric Equivalent or Enumerated Type: Specifies a binding edge.

- 0 LeftEdge!
- 1 RightEdge!
- 2 TopEdge!
- 3 BottomEdge!

*Offset*

Measurement Expression: Specifies the binding offset. The measurement will be added to the margin of the edge specified in the Edge parameter.

---

|                 |                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Page, Binding/Duplex, select a binding width option, specify an amount.                                      |
| <b>Example</b>  | To set a binding offset of 0.75" at the top of a document, the command is:<br>Binding (Edge: TopEdge; Offset: 0.75") |
| <b>See Also</b> | ♦ ?Binding                                                                                                           |

---

## BindingOptionsDlg

---

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Purpose</b> | BindingOptionsDlg displays the Binding Options dialog box. |
| <b>Syntax</b>  | BindingOptionsDlg ()                                       |
| <b>Route</b>   | Layout, Page, Binding.                                     |

---

## BlockProtect

---

|                   |                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BlockProtect turns Block Protect on or off.                                                         |
| <b>Syntax</b>     | BlockProtect (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                          |
| <b>Parameters</b> |                                                                                                     |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether Block Protect is on.<br>0 Off!<br>1 On!    |
| <b>Route</b>      | Layout, Page, Keep Text Together, select or deselect the Keep Text Together on Same Page check box. |
| <b>Example</b>    | To turn on Block Protect, the command is:<br>BlockProtect (State: On!)                              |
| <b>See Also</b>   | ♦ ?BlockProtect                                                                                     |

---

---

## BoldKey

|                 |                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | BoldKey toggles Bold at the insertion point.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | BoldKey ()                                                                                                                                       |
| <b>See Also</b> | ♦ AttributeAppearanceToggle                                                                                                                      |

---

## BookmarkBlock

|                   |                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BookmarkBlock executes GoTo & Select, which finds and selects a bookmark with assigned text.                              |
| <b>Syntax</b>     | BookmarkBlock (Name: <i>Character Expression</i> )                                                                        |
| <b>Parameters</b> |                                                                                                                           |
| <i>Name</i>       | Character Expression: Specifies the name of a bookmark with assigned text.                                                |
| <b>Route</b>      | Insert, Bookmark, select a bookmark, Go To & Select.                                                                      |
| <b>Example</b>    | To find a bookmark named Select This and select the assigned text, the command is:<br>BookmarkBlock (Name: "Select This") |
| <b>See Also</b>   | ♦ BookmarkCreate ♦ BookmarkDelete ♦ BookmarkDlg ♦ BookmarkFind<br>♦ BookmarkMove ♦ BookmarkRename                         |

---

## BookmarkCreate

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <b>Purpose</b> | BookmarkCreate creates a bookmark.                          |
| <b>Syntax</b>  | BookmarkCreate (BookmarkName: <i>Character Expression</i> ) |

---

**Parameters**

*Name* Character Expression: Specifies the name of a bookmark to create.

---

**Route** Insert, Bookmark, Create, enter a bookmark name.

---

**Example** To create a bookmark named Find Me, the command is:

BookmarkCreate (Name: "Find Me")

---

**See Also** ♦ BookmarkBlock ♦ BookmarkDelete ♦ BookmarkDlg ♦ BookmarkFind  
♦ BookmarkMove ♦ BookmarkRename

---

## BookmarkDelete

---

**Purpose** BookmarkDelete deletes a bookmark.

---

**Syntax** BookmarkDelete (Name: *Character Expression*)

---

**Parameters**

*Name* Character Expression: Specifies the name of a bookmark to delete.

---

**Route** Insert, Bookmark, select a bookmark, Delete.

---

**Example** To delete a bookmark named Last Page, the command is:

BookmarkDelete (Name: "Last Page")

---

**See Also** ♦ BookmarkBlock ♦ BookmarkCreate ♦ BookmarkDlg ♦ BookmarkFind  
♦ BookmarkMove ♦ BookmarkRename

---

## BookmarkDlg

---

**Purpose** BookmarkDlg displays the Bookmark dialog box.

---

**Syntax** BookmarkDlg ()

---

**Route** Insert, Bookmark, Go To.



---

**See Also**

- ♦ BookmarkBlock ♦ BookmarkCreate ♦ BookmarkDelete ♦ BookmarkFind
- ♦ BookmarkMove ♦ BookmarkRename

---

## BookmarkFind

---

**Purpose**

BookmarkFind moves the insertion point to a specified bookmark.

---

**Syntax**

BookmarkFind (Name: *Character Expression*)

---

**Parameters***Name*

Character Expression: Specifies the name of a bookmark to find.

---

**Route**

Insert, Bookmark, select a bookmark, Go To.

---

**Example**

To find a bookmark named Entry, the command is:

```
BookmarkFind (Name: "Entry")
```

---

**See Also**

- ♦ BookmarkBlock ♦ BookmarkCreate ♦ BookmarkDelete ♦ BookmarkDlg
- ♦ BookmarkMove ♦ BookmarkRename

---

## BookmarkMove

---

**Purpose**

BookmarkMove moves a bookmark from its current location to the insertion point.

---

**Syntax**

BookmarkMove (Name: *Character Expression*)

---

**Parameters***Name*

Character Expression: Specifies a bookmark to move to the insertion point.

---

**Route**

Insert, Bookmark, select a bookmark, Move.

---

**Example**

To move a bookmark named Read This, the command is:

```
BookmarkMove (Name: "Read This")
```

---

**See Also**

- ♦ BookmarkBlock ♦ BookmarkCreate ♦ BookmarkDelete ♦ BookmarkDlg
- ♦ BookmarkFind ♦ BookmarkRename

---

# BookmarkRename

---

|                   |                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BookmarkRename renames a bookmark. If you type in the name of a bookmark that already exists in the current document you will receive an error message. |
| <b>Syntax</b>     | BookmarkRename (OldName: <i>Character Expression</i> ; NewName: <i>Character Expression</i> )                                                           |
| <b>Parameters</b> |                                                                                                                                                         |
| <i>OldName</i>    | Character Expression: Specifies the name of a bookmark to rename.                                                                                       |
| <i>NewName</i>    | Character Expression: Specifies the new name of a bookmark.                                                                                             |
| <b>Route</b>      | Insert, Bookmark, select a bookmark, Rename, enter a new name.                                                                                          |
| <b>Example</b>    | To rename a bookmark named Edit as Copy, the command is:<br><br>BookmarkRename (OldName: "Edit"; NewName: "Copy")                                       |
| <b>See Also</b>   | ♦ BookmarkBlock ♦ BookmarkCreate ♦ BookmarkDelete ♦ BookmarkDlg<br>♦ BookmarkFind ♦ BookmarkMove                                                        |

---

---

# BorderBottomLine

---

|                   |                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BorderBottomLine specifies the style of a bottom border line.<br><br>IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleEdit, and be followed by a closing command, such as BorderStyleEnd.                                                                                                                 |
| <b>Syntax</b>     | BorderBottomLine (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                              |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                             |
| <i>Style</i>      | Numeric Equivalent or Enumerated Type: Specifies a bottom border line style. To specify a user-defined style, enclose the name in quotation marks.<br><br>0 SingleLine!<br>1 DoubleLine!<br>2 DashedLine!<br>3 DottedLine!<br>4 ThickLine!<br>5 ExtraThickLine!<br>6 ThinThickLine!<br>7 ThickThinLine!<br>8 ButtonTopLeftLine!<br>9 ButtonBottomRightLine! |

126 TableDefaultLine!  
127 NoLine!

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, deselect All, select Bottom, select a line style.

---

**Example** To create a paragraph border with a double bottom border line, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)
BorderBottomLine (Style: DoubleLine!)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ BorderLeftLine ♦ BorderRightLine ♦ BorderSeparatorLine ♦ BorderTopLine  
♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderBottomLine

---

## BorderColor

---

**Purpose** BorderColor specifies the color of a paragraph, page, or column border. All parameters for this command are optional.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate and BorderUseBorderColor and be followed by a closing command such as BorderStyleEnd.

IMPORTANT: All parameters for this command are optional. If you do not fill in a particular parameter, the current program setting for that aspect of the feature will be used when the macro plays.

---

**Syntax** BorderColor (ColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; Shade: *Numeric Expression*)

---

### Parameters

*ColorName* Character Expression (optional): Specifies a border color. Includes custom-made colors.

*Red* Numeric Expression (optional): Specifies the amount of red in a selected color. Values range from 0 to 255. The values of a selected color are available in the Color Selection dialog box.

*Green* Numeric Expression (optional): Specifies the amount of green in the selected color. Values range from 0 to 255. The values of a selected color are available in the Color Selection dialog box.

|                 |                                                                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Blue</i>     | Numeric Expression (optional): Specifies the amount of blue in the selected color. Value range from 0 to 255. The values of a selected color are available in the Color Selection dialog box.                                                                                           |
| <i>Shade</i>    | Numeric Expression (optional): Specifies the shading percentage of the selected color. This value can range from 0 to 100.                                                                                                                                                              |
| <b>Route</b>    | Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, select a border color.                                                                                                                                                                                |
| <b>Example</b>  | To create a border style with a magenta border, the command sequence is:<br><br><pre> BorderStyleEdit (BorderName: SingleBorder!) BorderUseBorderColor (State: Yes!) BorderColor (ColorName: "Magenta"; Red: 255; Green: 0; Blue: 255; Shade: 100) BorderStyleEnd (State: Save!) </pre> |
| <b>See Also</b> | <ul style="list-style-type: none"> <li>♦ BorderStyleCreate ♦ BorderStyleEdit ♦ BorderStyleEnd ♦ BorderUseBorderColor</li> <li>♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderColorBlue</li> <li>♦ ?BorderColorGreen ♦ ?BorderColorRed ♦ ?BorderColorShading</li> </ul>    |

---

## BorderCornerRadius

|                   |                                                                                                                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BorderCornerRadius customizes the curve of rounded corners.<br><br><b>IMPORTANT:</b> This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command, such as BorderStyleEnd. |
| <b>Syntax</b>     | BorderCornerRadius (Radius: <i>Measurement Expression</i> )                                                                                                                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                                                                                 |
| <i>Radius</i>     | Measurement Expression: Specifies the curve of border corners. Higher measurements produce more rounded corners.                                                                                                                                |
| <b>Route</b>      | Graphics, Graphics Styles, Border, Create (or select a border style, Edit), specify a radius in the Corners entry box.                                                                                                                          |
| <b>Example</b>    | To create a border style named My Style with a border corner radius of 0.5", the command sequence is:<br><br><pre> BorderStyleCreate (BorderName: "My Style"; ) BorderCornerRadius (Radius: 0.5") BorderStyleEnd (State: Save!) </pre>          |

---

**See Also**

- ♦ BorderStyleCreate ♦ BorderStyleEdit ♦ BorderStyleEnd ♦ TextBorderCreate
- ♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderCornerRadius

---

## BorderDropShadow

---

**Purpose**

BorderDropShadow specifies the location, size, and color of a border shadow.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and be followed by a closing command, such as BorderStyleEnd.

---

**Syntax**

BorderDropShadow (Location: *Numeric Equivalent or Enumerated Type*; Size: *Measurement Expression*; ColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; Shade: *Numeric Expression*)

---

**Parameters**

|                  |                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Location</i>  | Numeric Equivalent or Enumerated Type: Specifies the location of a shadow.<br><ul style="list-style-type: none"><li>0 NoShadow!</li><li>1 UpperLeft!</li><li>2 LowerLeft!</li><li>3 LowerRight!</li><li>4 UpperRight!</li></ul> |
| <i>Size</i>      | Measurement Expression (optional): Specifies the size of a shadow. The default shadow size is .125".                                                                                                                            |
| <i>ColorName</i> | Numeric Expression (optional): Specifies a shadow color, including custom-made colors.                                                                                                                                          |
| <i>Red</i>       | Numeric Expression (optional): Specifies the amount of red in a selected color. Values range from 0 to 255 and are available in the Color Selection dialog box.                                                                 |
| <i>Green</i>     | Numeric Expression (optional): Specifies the amount of green in a selected color. Values range from 0 to 255 and are available in the Color Selection dialog box.                                                               |
| <i>Blue</i>      | Numeric Expression (optional): Specifies the amount of blue in a selected color. Values range from 0 to 255 and are available in the Color Selection dialog box.                                                                |
| <i>Shade</i>     | Numeric Expression (optional): Specifies the shading percentage of a selected color. Values range from 0 to 100.                                                                                                                |

---

**Route**

Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, select a drop shadow type, color, and width.

---

**Example**

To create a border style with a blue shadow on the upper right corner of a border, the command sequence is:

```
BorderStyleCreate (BorderName: SingleLine!)
BorderDropShadow (Location: UpperRight!, Size: 0.125"; ColorName: "Blue"; Red: 0; Green: 0; Blue:
255; Shade: 100)
BorderStyleEnd (State: Save!)
```

---

**See Also**

• BorderStyleCreate • BorderStyleEdit • BorderStyleEnd

---

## BorderFillColors

---

**Purpose**

BorderFillColors specifies the color of a border fill.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command such as BorderStyleEnd.

---

**Syntax**

BorderFillColors (ForeColorName: *Character Expression*; ForeRed: *Numeric Expression*; ForeGreen: *Numeric Expression*; ForeBlue: *Numeric Expression*; ForeShade: *Numeric Expression*; BackColorName: *Character Expression*; BackRed: *Numeric Expression*; BackGreen: *Numeric Expression*; BackBlue: *Numeric Expression*; BackShade: *Numeric Expression*)

---

**Parameters**

IMPORTANT: The first five parameters specify foreground color and the last five specify background color. Select a background color only if you have already selected a fill pattern.

*ForeColorName*

Character Expression (optional): Specifies a border fill color, including custom-made colors.

*ForeRed*

Numeric Expression (optional): Specifies the amount of red in a selected color. Values range from 0 to 255 and are available in the Color Selection dialog box.

*ForeGreen*

Numeric Expression (optional): Specifies the amount of green in a selected color. Values range from 0 to 255 and are available in the Color Selection dialog box.

*ForeBlue*

Numeric Expression (optional): Specifies the amount of blue in a selected color. Values range from 0 to 255 and are available in the Color Selection dialog box.

*ForeShade*

Numeric Expression (optional): Specifies the shading percentage of a selected color. Values range from 0 to 100.

*BackColorName*

Character Expression (optional): Specifies a border fill background color, including custom-made colors.

|                  |                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>BackRed</i>   | Numeric Expression (optional): Specifies the amount of red in a selected background color. Values range from 0 to 255 and are available in the Color Selection dialog box.   |
| <i>BackGreen</i> | Numeric Expression (optional): Specifies the amount of green in a selected background color. Values range from 0 to 255 and are available in the Color Selection dialog box. |
| <i>BackBlue</i>  | Numeric Expression (optional): Specifies the amount of blue in a selected background color. Values range from 0 to 255 and are available in the Color Selection dialog box.  |
| <i>ForeShade</i> | Numeric Expression (optional): Specifies the shading percentage of a selected background color. Values range from 0 to 100.                                                  |

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, select options, Palette, select options.

---

**Example** To create a 40% shaded fill paragraph border with a red foreground, the command sequence is:

```
TextBorderCreate (Style: SingleLine!, FillName: Fill40!, BorderFunction: ParagraphBorder!)
BorderFillColors (ForeColorName: "Red"; ForeRed: 255; ForeGreen: 0; ForeBlue: 0; ForeShade: 40)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ FillColors ♦ FillSetWPG2 ♦ TextBorderCreate ♦ TextBorderEdit  
♦ TextBorderEnd ♦ TextBorderFillStyle

---

## BorderInsideSpacing

**Purpose** BorderInsideSpacing specifies spacing inside a border.

IMPORTANT: This option is not available with page borders.

IMPORTANT: This command must be preceded by a command which specifies a border, such as TextBorderCreate, and followed by a closing command such as TextBorderEnd.

---

**Syntax** BorderInsideSpacing (Bottom: *Measurement Expression*; Left: *Measurement Expression*; Right: *Measurement Expression*; Top: *Measurement Expression*)

---

## Parameters

|               |                                                                 |
|---------------|-----------------------------------------------------------------|
| <i>Bottom</i> | Measurement Expression: Specifies bottom inside border spacing. |
| <i>Left</i>   | Measurement Expression: Specifies left inside border spacing.   |
| <i>Right</i>  | Measurement Expression: Specifies right inside border spacing.  |
| <i>Top</i>    | Measurement Expression: Specifies top inside border spacing.    |

---

## Route

Graphics, Graphics Styles, Border, Create (or select a border style, Edit), select sides to modify, specify an inside space measurement.

---

## Example

To create inside spacing of .1" all the way around a paragraph border, the command is:

```
TextBorderCreate (Style: "New Border"; Fillname: NoFill!; BorderFunction: ParagraphBorder!)
BorderInsideSpacing (Bottom: 0.1"; Left: 0.1"; Right: 0.1"; Top: 0.1")
TextBorderEnd (State: Save!)
```

---

## See Also

- ♦ BorderOutsideSpacing ♦ BorderStyleCreate ♦ BorderStyleEdit ♦ BorderStyleEnd
- ♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd
- ♦ ?BorderInsideSpacingTop ♦ ?BorderInsideSpacingBottom

---

# BorderLeftLine

---

## Purpose

BorderLeftLine specifies the left line style of a border.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command such as BorderStyleEnd.

---

## Syntax

BorderLeftLine (Style: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

### *Style*

Numeric Equivalent or Enumerated Type: Specifies a left border line style. To specify a user-defined line style, enclose the name in quotation marks.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!



9 ButtonBottomRightLine!  
126 TableDefaultLine!  
127 NoLine!

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, deselect All, select Left, select a Line style.

---

**Example** To create a thick left paragraph border line, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)
BorderLeftLine (Style: ThickLine!)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ BorderBottomLine ♦ BorderRightLine ♦ BorderSeparatorLine ♦ BorderTopLine  
♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderLeftLine

---

## BorderOutsideSpacing

---

**Purpose** BorderOutsideSpacing specifies spacing outside a border.  
IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command such as BorderStyleEnd.

---

**Syntax** BorderOutsideSpacing (Left: *Measurement Expression*; Right: *Measurement Expression*;  
Top: *Measurement Expression*; Bottom: *Measurement Expression*)

---

### Parameters

*Left* Measurement Expression (optional): Specifies left outside border spacing.  
*Right* Measurement Expression (optional): Specifies right outside border spacing.  
*Top* Measurement Expression (optional): Specifies top outside border spacing.  
*Bottom* Measurement Expression (optional): Specifies bottom outside border spacing.

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, specify an outside space measurement.

---

**Example** To create a shadow border style with outside spacing of .2" on all sides of a border, the command sequence is:

```
BorderStyleCreate (BorderName: "Shadow")
BorderOutsideSpacing (Left: .2"; Right: .2"; Top: .2"; Bottom: .2")
BorderStyleEnd (State: Save!)
```

---

**See Also**

♦ [BorderInsideSpacing](#) ♦ [TextBorderCreate](#) ♦ [TextBorderEdit](#) ♦ [TextBorderEnd](#)

---

## BorderRightLine

---

**Purpose**

`BorderRightLine` specifies the right line border style.

**IMPORTANT:** This command must be preceded by a command which specifies a border, such as `BorderStyleCreate`, and followed by a closing command such as `BorderStyleEnd`.

---

**Syntax**

`BorderRightLine` (Style: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Style*

Numeric Equivalent or Enumerated Type: Specifies a right border line style. To specify a user-defined line style, enclose the name in quotation marks.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!
- 9 ButtonBottomRightLine!
- 126 TableDefaultLine!
- 127 NoLine!

---

**Route**

Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, deselect All, select Right, select a line style.

---

**Example**

To create a Paragraph border with a dashed right line, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)
BorderRightLine (Style: DashedLine!)
TextBorderEnd (State: Save!)
```

---

**See Also**

♦ [BorderBottomLine](#) ♦ [BorderLeftLine](#) ♦ [BorderSeparatorLine](#) ♦ [BorderTopLine](#)  
♦ [TextBorderCreate](#) ♦ [TextBorderEdit](#) ♦ [?BorderRightLine](#)

---

# BorderSeparatorLine

---

**Purpose** BorderSeparatorLine specifies the separator line style in a paragraph or column border.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command such as BorderStyleEnd.

---

**Syntax** BorderSeparatorLine (Style: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies the type of separator line a border contains. To specify a user-defined line style, enclose the name in quotation marks.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!
- 9 ButtonBottomRightLine!
- 126 TableDefaultLine!
- 127 NoLine!

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, deselect All, select Separator, select a line style.

---

**Example** To create a double-line separator in a paragraph border, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)
BorderSeparatorLine (Style: DoubleLine!)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ BorderBottomLine ♦ BorderLeftLine ♦ BorderRightLine ♦ BorderTopLine  
♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderSeparatorLine

---

# BorderStyleSpacing

|                   |                                                                                                                                                                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BorderStyleSpacing specifies whether the Automatic Spacing option is selected. If not selected, you can specify a border's inside and/or outside spacing.<br><br>IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command such as BorderStyleEnd. |
| <b>Syntax</b>     | BorderStyleSpacing (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                      |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                       |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether the Automatic Spacing option is selected.<br><br>0 No!<br>1 Yes!                                                                                                                                                                                                             |
| <b>Route</b>      | Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, Auto Spacing.                                                                                                                                                                                                                                       |
| <b>Example</b>    | To specify that Automatic Spacing is selected, the command sequence is:<br><br>TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)<br>BorderStyleSpacing (State: No!)<br>TextBorderEnd (State: Save!)                                                                                        |
| <b>See Also</b>   | ♦ BorderInsideSpacing ♦ BorderOutsideSpacing ♦ TextBorderCreate<br>♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderStyleSpacing                                                                                                                                                                                                             |

---

# BorderStyleCopy

|                |                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BorderStyleCopy copies a specified style from one location to another.                                                                                                                                                                                    |
| <b>Syntax</b>  | BorderStyleCopy (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , FromLibrary: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , ToLibrary: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , DestName: <i>Character Expression</i> ) |

---

## Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i>       | Numeric Equivalent or Enumerated Type: Specifies the name of a style to copy. To specify a user-defined style, enclose the name in quotation marks.<br><br>0 SpacingOnly!<br>1 SingleBorder!<br>2 DoubleBorder!<br>3 DashedBorder!<br>4 DottedBorder!<br>5 ThickBorder!<br>6 ExtraThickBorder!<br>7 ThinThickBorder!<br>8 ThickThinBorder!<br>9 ThickTopBottomBorder!<br>10 ButtonBorder!<br>11 ColumnBorderBetween!<br>12 ColumnBorderAll!<br>127 NoBorder! |
| <i>FromLibrary</i> | Numeric Equivalent or Enumerated Type: Specifies location of a style to copy.<br><br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                                                                                                                                                                                                                                                                                                 |
| <i>ToLibrary</i>   | Numeric Equivalent or Enumerated Type: Specifies the destination of a style to copy.<br><br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                                                                                                                                                                                                                                                                                          |
| <i>DestName</i>    | Character Expression (optional): Renames a style to copy. Select this parameter only when placing a second copy of the style in the same location (when contents of Current Location and Destination are the same).                                                                                                                                                                                                                                          |

---

**Route** Graphics, Graphics Styles, Border, select a style, Options, Copy, enter destination name.

---

**Example** To copy a style named MyStyle from a personal style library to a shared style library, the command is:

BorderStyleCopy (Style: "MyStyle"; ToLibrary: PersonalLibrary!; DestLibrary: SharedLibrary!)

---

**See Also** ♦ BorderStyleRetrieve ♦ BorderStyleSave

---

# BorderStyleCreate

---

**Purpose** BorderStyleCreate names a new border style.

**Syntax** BorderStyleCreate (Style: *Character Expression*; Library: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*Style* Character Expression: Names a new style.

*Library* Numeric Equivalent or Enumerated Type (optional): Specifies a location for the new style.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Graphics, Graphics Styles, Border, Create, enter a Style name.

---

**Example** To create a border style named Highlight in the Personal Library, the command is:

BorderStyleCreate (Style: "Highlight"; Library: PersonalLibrary!)

---

**See Also** ♦ BorderStyleEnd ♦ BorderStyleEdit ♦ BorderStyleDelete ♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd

---

# BorderStyleDelete

---

**Purpose** BorderStyleDelete deletes a specified style in a current document or a personal or shared style library.

**Syntax** BorderStyleDelete (Style: *Numeric Equivalent or Enumerated Type*; Library: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies the name of a style to delete. To specify a user-defined style, enclose the name in quotation marks.

- 0 SpacingOnly!
- 1 SingleBorder!
- 2 DoubleBorder!
- 3 DashedBorder!
- 4 DottedBorder!
- 5 ThickBorder!
- 6 ExtraThickBorder!

- 7 ThinThickBorder!
- 8 ThickThinBorder!
- 9 ThickTopBottomBorder!
- 10 ButtonBorder!
- 11 ColumnBorderBetween!
- 12 ColumnBorderAll!
- 127 NoBorder!

*Library* Numeric Equivalent or Enumerated Type: Specifies the location of the style to delete.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Graphics, Graphic Styles, Border, select a style from the Styles list box, Options, Delete.

---

**Example** To delete a border style named Example in the current document, the command is:

BorderStyleDelete (Style: "Example"; Library: CurrentDoc!)

---

**See Also** ♦ BorderStyleCopy ♦ BorderStyleRetrieve ♦ BorderStyleSave ♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd

---

## BorderStyleEdit

---

**Purpose** BorderStyleEdit specifies a style to edit. It is usually followed by border modification commands such as BorderCornerRadius.

---

**Syntax** BorderStyleEdit (Style: *Numeric Equivalent or Enumerated Type*; Library: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

*Style* Numeric Equivalent or Enumerated Type: Names a style to edit. To edit a user-defined style, enclose the name in quotation marks.

- 0 SpacingOnly!
- 1 SingleBorder!
- 2 DoubleBorder!
- 3 DashedBorder!
- 4 DottedBorder!
- 5 ThickBorder!
- 6 ExtraThickBorder!
- 7 ThinThickBorder!
- 8 ThickThinBorder!
- 9 ThickTopBottomBorder!

- 10 ButtonBorder!
- 11 ColumnBorderBetween!
- 12 ColumnBorderAll!
- 127 NoBorder!

*Library*

Numeric Equivalent or Enumerated Type: Specifies the location of the style to edit.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit.

---

**Example** To edit the Thin Thick border line style, the command is:

BorderStyleEdit (Style: ThinThickBorder!)

---

**See Also** ♦ BorderStyleCreate ♦ BorderStyleEnd ♦ TextBorderCreate ♦ TextBorderEdit  
♦ TextBorderEnd ♦ ?Border

---

## BorderStyleEnd

---

**Purpose** BorderStyleEnd indicates the end of editing or creating a border style and specifies whether to save the changes.

---

**Syntax** BorderStyleEnd (State: *Numeric Equivalent or Enumerated Type*)

**Parameters**

*State* Numeric Equivalent or Enumerated Type: Specifies whether the most recent changes made to a style should be saved.

- 0 Cancel!
- 1 Save!

---

**Route** OK or Cancel when closing the Edit Border Style or Create Border Style dialog box.

---

**Example** To save the changes made to a border style, the command is:

BorderStyleEnd (State: Save!)



---

# BorderStyleName

---

**Purpose** BorderStyleName renames a border style.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

IMPORTANT: This command must be preceded by BorderStyleEdit or BorderStyleCreate and followed by BorderStyleEnd.

---

**Syntax** BorderStyleName (Style: *Character Expression*)

---

**Parameters**

*Style* Character Expression: Specifies the name of a border to rename.

---

**Example** To rename a border style from Nouveau to Traditional, the command sequence is:

```
BorderStyleEdit (Style: "Nouveau")
BorderStyleName (Style: "Traditional")
BorderStyleEnd (State: Save!)
```

---

# BorderStyleRetrieve

---

**Purpose** BorderStyleRetrieve retrieves all border styles in a specified style library into the current document or a personal or shared style library.

---

**Syntax** BorderStyleRetrieve (Filename: *Character Expression*; FromLibrary: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Filename* Character Expression: Specifies a style library containing styles to retrieve. Specify both the filename and path of the library.

*FromLibrary* Numeric Equivalent or Enumerated Type: Specifies where to place the retrieved styles.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Graphics, Graphics Styles, Border, Options, Retrieve, enter a filename.

---

**Example** To retrieve styles contained in a style library named Format into a personal style library, the command is:

BorderStyleRetrieve (Filename: "Format"; FromLibrary: PersonalLibrary!)

---

**See Also** ♦ StyleCopy ♦ BorderStyleSave

---

## BorderStyleSave

---

**Purpose** BorderStyleSave saves border styles in a current document or in a personal or shared style library to a specified style library.

IMPORTANT: Saving to an existing library deletes any styles it contains and replaces them with the new styles.

---

**Syntax** BorderStyleSave (Filename: *Character Expression*; Library: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

*Filename* Character Expression: Specifies the name of the style library to which the styles should be saved. Specify both the filename and path of the library.

*Library* Numeric Equivalent or Enumerated Type: Specifies the location of the styles to save.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Graphics, Graphics Styles, Border, Options, Save As.

---

**Example** To save all styles from the Shared Style Library to a style library named Designs, the command is:

BorderStyleSave (Filename: "Designs"; Library: SharedLibrary!)

---

**See Also** ♦ BorderStyleCopy ♦ BorderStyleRetrieve

---

## BorderTopLine

---

**Purpose** BorderTopLine specifies the top line style of a border.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and be followed by a closing command such as BorderStyleEnd.

---

---

**Syntax** BorderTopLine (Style: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Style* Numeric Equivalent or Enumerated Type: Specifies a top border line style. To specify a user-defined style, enclose the name in quotation marks.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!
- 9 ButtonBottomRightLine!
- 126 TableDefaultLine!
- 127 NoLine!

---

**Route** Graphics, Graphics Styles, Borders, select a style from the Styles list box, Edit, deselect All, select Top, select a line style.

---

**Example** To create a paragraph border with an extra thick top line, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)
BorderTopLine (Style: ExtraThickLine!)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ BorderBottomLine ♦ BorderLeftLine ♦ BorderRightLine ♦ BorderSeparatorLine  
♦ TextBorderCreate ♦ ?BorderTopLine

---

---

## BorderUseBorderColor

---

**Purpose** BorderUseBorderColor specifies whether the border color option is for the current border, and is used with BorderStyleCreate or BorderStyleEdit and BorderColor.

IMPORTANT: This command must be preceded by a command which specifies a border, such as BorderStyleCreate, and followed by a closing command such as BorderStyleEnd.

---

**Syntax** BorderUseBorderColor (State: *Numeric Equivalent* or *Enumerated Type*)

---

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to use the border color option.

- 0 No!
- 1 Yes!

---

**Route** Graphics, Graphics Styles, Border, select a style from the Styles list box, Edit, Border Color, select a color from the palette.

---

**Example** To use the color border option for the current paragraph border, the command is:

```
TextBorderCreate (Style: SingleBorder!; FillName: NoFill!; BorderFunction: ParagraphBorder!)
BorderUseBorderColor (State: Yes!)
BorderColor (ColorName: "Magenta"; Red: 255; Green: 0; Blue: 255; Shade: 100)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ BorderColor ♦ BorderStyleCreate ♦ BorderStyleEdit ♦ BorderStyleEnd  
♦ TextBorderCreate ♦ TextBorderEdit ♦ TextBorderEnd ♦ ?BorderUseBorderColor

---

## BoxAttachTo

---

**Purpose** BoxAttachTo attaches a graphics box to a page, paragraph, or character. When a box is attached to a location, BoxHorizontalPosition and BoxVerticalPosition specify horizontal and vertical positions.

IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxAttachTo (Anchor: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*Anchor* Numeric Equivalent or Enumerated Type: Specifies where a box is attached.

- 2 Page!
- 3 Paragraph!
- 64 Character!

---

**Route** Graphics, choose a box type, Position, select a Box Placement option.

---

**Example** To attach a text box to a character, the command sequence is:

```
BoxCreate (BoxType: TextBox!)
BoxAttachTo (Anchor: Character!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxHorizontalPosition ♦ BoxVerticalPosition  
♦ ?BoxAttachTo

---

## BoxBorder

---

**Purpose** BoxBorder specifies a box border style.  
IMPORTANT: This command must be preceded by BoxCreate or BoxEdit and followed by BoxEnd.

---

**Syntax** BoxBorder (Style: *Character Expression*)

---

### Parameters

*Style* Character Expression: Specifies the name of a box border style.

---

**Route** Graphics, Edit Box, Border/Fill, select a border style.

---

**Example** To specify a hairline border for Box 1, the command sequence is:

```
BoxEdit (BoxNumber: 1)
BoxBorder (Style: "Hairline")
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd

---

## BoxBorderDlg

---

**Purpose** BoxBorderDlg displays the Box Border/Fill dialog box.

---

**Syntax** BoxBorderDlg ()

---

**Route** Graphics, Graphics Styles, Box, select a style, Edit, Border/Fill.

---

## BoxCaptionEdit

---

**Purpose** BoxCaptionEdit opens a box caption.  
IMPORTANT: Because this command has no parameters, it is always followed by commands such as Type that furnish caption text.

---

|               |                                    |
|---------------|------------------------------------|
| <b>Syntax</b> | BoxCaptionEdit ()                  |
| <b>Route</b>  | Graphics, Edit Box, Caption, Edit. |

---

## BoxCaptionEditEnd

---

|                |                                                                                                                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxCaptionEditEnd closes a box caption.<br><br>IMPORTANT: This command must be preceded by BoxCaptionEdit and commands that furnish or modify caption text, such as Type.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

|               |                      |
|---------------|----------------------|
| <b>Syntax</b> | BoxCaptionEditEnd () |
|---------------|----------------------|

---

---

|                 |                                     |
|-----------------|-------------------------------------|
| <b>See Also</b> | ♦ BoxCaptionEdit ♦ SubstructureExit |
|-----------------|-------------------------------------|

---

## BoxCaptionFormatting

---

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxCaptionFormatting specifies the width of a box caption.<br><br>IMPORTANT: This command must be preceded by BoxCreate or BoxEdit and followed by BoxEnd. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

|               |                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | BoxCaptionFormatting ( <i>SizeAuto: Numeric Equivalent or Enumerated Type, or Measurement Expression, or Numeric Expression</i> ) |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|

---

### Parameters

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>SizeAuto</i> | Numeric Equivalent or Enumerated Type: Specifies whether a box caption width is set automatically, is fixed, or is a percentage of a box width. To base the width on a current unit of measure, use a measurement expression. To base the width on a percentage of the box width, use a numeric expression. To set the width automatically, use a numeric equivalent or an enumerated type. |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

2 AutoWidth!

---

|              |                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Graphics, Custom Box or Edit Box, select a style name, OK, Caption, select Auto Width or specify fixed or percent amount. |
|--------------|---------------------------------------------------------------------------------------------------------------------------|

---

---

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| <b>Example</b> | To specify a caption width of one inch for box 1, the command sequence is: |
|----------------|----------------------------------------------------------------------------|

---

```
BoxEdit (BoxNumber: 1)
BoxCaptionFormatting (SizeAuto: 1")
BoxEnd (State: Save!)
```

---

## BoxCaptionInitialStyle

---

**Purpose** BoxCaptionInitialStyle specifies an initial style for a box caption.  
IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

**Syntax** BoxCaptionInitialStyle (Style: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies the initial style for a box caption.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!

45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

---

**Route** Graphics, Graphics Styles, Box, select a style, Edit, Caption, choose the Change button for Initial Style, select a style, Apply.

---

**Example** To specify Comment Style as the caption initial style, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxCaptionInitialStyle (Style: CommentStyle!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxCaptionDefaultInitialStyle



---

# BoxCaptionNumberStyle

---

**Purpose** BoxCaptionNumberStyle specifies a numbering style for box caption.  
IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxCaptionNumberStyle (Style: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies a number style for a box caption.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutInStyle!
- 46 OutlineStyle!
- 47 LegalOutInStyle!
- 48 BulletsOutInStyle!
- 49 HeadingsOutInStyle!

50 Legal\_2OutInStyle!  
51 NumbersOutInStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

---

**Route** Graphics, Graphics Styles, select Box, select a style, Edit, Caption, choose the Change button for Number Style, select a style, Apply.

---

**Example** To specify Bullet Style as the caption numbering style, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxCaptionNumberStyle (Style: BulletStyle!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxCaptionDefaultNumberStyle

---

## BoxCaptionPosition

---

**Purpose** BoxCaptionPosition customizes the location of a graphics box caption.

IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

**Syntax**                      *BoxCaptionPosition* (Side: *Numeric Equivalent* or *Enumerated Type*; Border: *Numeric Equivalent* or *Enumerated Type*; Alignment: *Numeric Equivalent* or *Enumerated Type*; AlignmentOffset: *Numeric Expression*)

---

**Parameters**

*Side*                              Numeric Equivalent or Enumerated Type: Specifies the side of the box where the caption is located.

- 0 Left!
- 1 Top!
- 3 Right!
- 4 Bottom!

*Border*                            Numeric Equivalent or Enumerated Type: Specifies a caption's location relative to the border.

- 0 OutsideBorder!
- 1 InsideBorder!
- 2 OnBorder!

*Alignment*                      Numeric Equivalent or Enumerated Type: Specifies the position of a caption.

- 0 Left!
- 1 Top!
- 2 Center!
- 3 Right!
- 4 Bottom!

*AlignmentOffset*              Numeric Expression: Specifies the position offset.

---

**Route**                            Graphics, choose a type, Caption, select options from Side of Box in Caption Position group box, Position, specify an absolute or percent offset.

---

**See Also**                        ♦ ?BoxCaptionPositionAlignment ♦ ?BoxCaptionPositionBorder  
♦ ?BoxCaptionPositionOffset

---

**Example**                        To position a caption within the top border of a graphics box, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxCaptionPosition (Side: Top!; Border: InsideBorder!; Alignment: Left!; Alignment offset: .5")
BoxEnd (State: Save!)
```

---

# BoxCaptionRotation

---

## Purpose

BoxCaptionRotation rotates graphics box captions.

IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and BoxUpdateDisplay, and be followed by a closing command such as BoxEnd.

---

## Syntax

BoxCaptionRotation (Rotation: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

### *Rotation*

Numeric Equivalent or Enumerated Type: Specifies the rotation degree of a graphics box caption.

- 0 None!
- 1 Degrees90!
- 2 Degrees180!
- 3 Degrees270!

---

## Route

Graphics, choose a box type, Caption button, select a rotation option.

---

## Example

To rotate a caption 90 degrees, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxUpdateDisplay ()
BoxCaptionRotation (Rotation: Degrees90!)
BoxEnd (State: Save!)
```

---

## See Also

♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxUpdateDisplay

---

# BoxChangeLineHeight

---

## Purpose

BoxChangeLineHeight specifies whether attaching a graphics box to a character position changes the text line height. The default changes the height.

IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

## Syntax

BoxChangeLineHeight (State: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether the graphics box changes the text line height when attached to a character position.

- 0 No!
- 1 Yes!

---

**Route** Graphics, choose a box type, Position, select Treat Box as Character and Box Changes Text Line Height.

---

**Example** To prevent the graphics box from changing the text line height, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxChangeLineHeight (State: No!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxChangeLineHeight

---

## BoxChangeStyle

---

**Purpose** BoxChangeStyle changes the style of a graphics box when creating or editing a box. IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxChangeStyle (BoxType: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*BoxType* Numeric Equivalent or Enumerated Type: Names the box style to change. To specify a user-defined style, enclose the name in quotation marks.

- 0 FigureBox!
- 1 TableBox!
- 2 TextBox!
- 3 UserBox!
- 4 EquationBox!
- 5 ButtonBox!
- 6 WatermarkImageBox!
- 7 InlineEquationBox!
- 127 NoBox!

|                 |                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Graphics, choose a box type, Style, select a style.                                                                                                                          |
| <b>Example</b>  | To change a box style from graphics to text, the command sequence is:<br><pre>BoxCreate (BoxType: FigureBox!) BoxChangeStyle (BoxType: TextBox!) BoxEnd (State: Save!)</pre> |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                               |

---

## BoxContentEdit

|                 |                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | BoxContentEdit creates and edits box contents.<br><b>IMPORTANT:</b> This command has no parameters and is always followed by commands that furnish box contents. |
| <b>Syntax</b>   | BoxContentEdit ()                                                                                                                                                |
| <b>Route</b>    | Graphics, choose a box type, Content, Edit.                                                                                                                      |
| <b>See Also</b> | ♦ BoxContentType                                                                                                                                                 |

---

## BoxContentPosition

|                   |                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxContentPosition positions graphics box contents.                                                                                                   |
| <b>Syntax</b>     | BoxContentPosition (Horizontal: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Vertical: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                                                                                       |
| <i>Horizontal</i> | Numeric Equivalent or Enumerated Type: Specifies the horizontal position of box contents.<br><pre>0 Left! 2 Center! 3 Right!</pre>                    |
| <i>Vertical</i>   | Numeric Equivalent or Enumerated Type: Specifies the vertical position of box contents.                                                               |

- 1 Top!
- 2 Center!
- 4 Bottom!

---

**Route** Graphics, choose a box type, Content, select a horizontal position and a vertical position.

---

**Example** To position contents at the left and bottom of a box, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxContentPosition (Horizontal: Left!; Vertical: Bottom!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxContentHorizontalPosition  
♦ ?BoxContentVerticalPosition

---

## BoxContentPreserveAspectRatio

---

**Purpose** BoxContentPreserveAspectRatio specifies whether to preserve the width/height ratio when the position of box contents changes.

IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxContentPreserveAspectRatio (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to preserve the image width/height ratio when the size of box contents changes.

- 0 No!
- 1 Yes!

---

**Route** Graphics, choose a box type, Content, Preserve Image Width/Height Ratio.

---

**Example** To preserve the image width/height ratio, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxContentPreserveAspectRatio (State: Yes!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxContentPreserveAspectRatio

---

# BoxContentType

---

## Purpose

BoxContentType specifies the content type of a graphics box.

IMPORTANT: This command must be preceded by a command which identifies a box, such as BoxCreate or BoxEdit, and followed by a closing command such as BoxEnd.

---

## Syntax

BoxContentType (Content: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

### Content

Numeric Equivalent or Enumerated Type: Specifies the type of contents in a graphics box.

- 1 Text!
- 2 LinkedText!
- 3 Image!
- 4 Equation!
- 5 Presentation!
- 6 Video!
- 7 Macro!
- 8 External!
- 127 Empty!

---

## Route

Graphics, choose a box type, Content, select a type from the Content pop-up list.

---

## Example

To specify box contents as a graphics image, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxContentType (Content: Image!)
BoxEnd (State: Save!)
```

---

## See Also

♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxContentType

---

# BoxCounter

---

## Purpose

BoxCounter assigns a new counter type to a current graphics box or graphics box style.

---

## Syntax

BoxCounter (CounterName: *Numeric Equivalent* or *Enumerated Type* or *Character Expression*)



---

## Parameters

*CounterName* Numeric Equivalent or Enumerated Type or Character Expression: Specifies the counter type of a current graphics box or graphics box style. To specify a user-defined counter type, use a character expression.

- 0 None!
- 1 FigureCounter!
- 2 TableBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

---

## Route

Graphics, Graphics Styles, Box, select a style, Create or Edit, Caption, change Counter in Caption Numbering and Styles group box.

---

## Example

To assign a Text Box counter type to a graphics box style, the command sequence is:

```
BoxStyleEdit (Style: FigureBox!; Library: PersonalLibrary!)
BoxCounter (CounterName: TextBoxCounter!)
```

---

## See Also

♦ [BoxCreate](#) ♦ [BoxEdit](#) ♦ [BoxEnd](#) ♦ [BoxEditByCounter](#) ♦ [?BoxCounter](#)

---

# BoxCreate

---

## Purpose

BoxCreate creates the specified graphics box.

---

## Syntax

BoxCreate (BoxType: *Numeric Equivalent or Enumerated Type*)

IMPORTANT: This command must be followed by BoxEnd.

---

## Parameters

*BoxType* Numeric Equivalent or Enumerated Type: Specifies the type of box to create.

- 0 FigureBox!
- 1 TableBox!
- 2 TextBox!
- 3 UserBox!
- 4 EquationBox!
- 5 ButtonBox!
- 6 WatermarkImageBox!
- 7 InlineEquationBox!
- 127 NoBox!

|                 |                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Graphics, Custom Box, select a box style from the Style Name list box.                                                        |
| <b>Example</b>  | To create an Equation Box, the command sequence is:<br><br><pre>BoxCreate (BoxType: EquationBox!) BoxEnd (State: Save!)</pre> |
| <b>See Also</b> | ♦ BoxEdit ♦ BoxEnd                                                                                                            |

---

## BoxDelete

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxDelete deletes a specified graphics box.<br><br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | BoxDelete (BoxType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; BoxNumber: <i>Numeric Expression</i> )                                           |

### Parameters

*BoxType*                      Numeric Equivalent or Enumerated Type: Specifies the type of box to delete.

- 0 FigureBox!
- 1 TableBox!
- 2 TextBox!
- 3 UserBox!
- 4 EquationBox!
- 5 ButtonBox!
- 6 WatermarkImageBox!
- 7 InlineEquationBox!
- 127 NoBox!

*BoxNumber*                    Numeric Expression: Specifies the box number to delete.

|                 |                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------|
| <b>Example</b>  | To delete Equation Box 3, the command is:<br><br><pre>BoxDelete (BoxType: EquationBox!; BoxNumber: 3)</pre> |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                              |

---

# BoxEdit

|                   |                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxEdit specifies a box number to edit.<br><br>IMPORTANT: This command must be followed by box editing commands and BoxEnd. |
| <b>Syntax</b>     | BoxEdit (BoxNumber: <i>Numeric Expression</i> )                                                                             |
| <b>Parameters</b> |                                                                                                                             |
| <i>BoxNumber</i>  | Numeric Expression: Specifies a box number to edit.                                                                         |
| <b>Route</b>      | Graphics, Edit Box, select Document Box Number, specify box number.                                                         |
| <b>Example</b>    | To edit graphics box 3 of a document, the command sequence is:<br><br>BoxEdit (BoxNumber: 3)<br>BoxEnd (State: Save!)       |
| <b>See Also</b>   | ♦ BoxCreate ♦ BoxEnd                                                                                                        |

---

# BoxEditByCounter

|                             |                                                                                                                                                                                               |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>              | BoxEditByCounter edits a box specified by counter type and number level.                                                                                                                      |
| <b>Syntax</b>               | BoxEditByCounter (CounterName: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Counter Level Number: <i>Numeric Expression</i> )                                                        |
| <b>Parameters</b>           |                                                                                                                                                                                               |
| <i>CounterName</i>          | Numeric Equivalent or Enumerated Type: Specifies a box type to edit.<br><br>0 None!<br>1 FigureCounter!<br>2 TableBoxCounter!<br>3 TextBoxCounter!<br>4 UserBoxCounter!<br>5 EquationCounter! |
| <i>Counter Level Number</i> | Numeric Expression: Specifies counter level number of a box. If the edited counter has more than one level, specify each of the levels by repeating this parameter.                           |
| <b>Route</b>                | In a document containing multiple boxes, choose Graphics, Edit Box, Counter Number, specify counter and number.                                                                               |

---

**Example**

To edit text box 2, the command is:

```
BoxEditByCounter (CounterName: TextBoxCounter!; Counter Level Number: 2)
```

---

**See Also**

♦ `BoxCreate` ♦ `BoxEdit` ♦ `BoxEnd`

---

---

## BoxEditNext

---

**Purpose**

`BoxEditNext` edits the next occurrence of a box type.

**IMPORTANT:** This command must be followed by box editing commands and `BoxEnd`.

---

**Syntax**

`BoxEditNext` (BoxType: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***BoxType*

Numeric Equivalent or Enumerated Type: Specifies a box type to edit. If a box type is not specified, the macro edits the box following the insertion point, regardless of type.

- 0 FigureBox!
  - 1 TableBox!
  - 2 TextBox!
  - 3 UserBox!
  - 4 EquationBox!
  - 5 ButtonBox!
  - 6 WatermarkImageBox!
  - 7 InlineEquationBox!
  - 127 NoBox!
- 

**Route**

With a box selected, Graphics, Edit Box, select Next from the Graphics feature bar.

---

**Example**

To edit the next occurrence of a user box, the command sequence is:

```
BoxEditNext (BoxType: UserBox!)
BoxEnd (State: Save!)
```

---

**See Also**

♦ `BoxEdit` ♦ `BoxEditByCounter` ♦ `BoxEditPrevious`

---

---

# BoxEditPrevious

---

|                   |                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxEditPrevious edits the previous occurrence of a box type.<br><br>IMPORTANT: This command must be followed by box editing commands and BoxEnd.                                                                                                                                                                                                             |
| <b>Syntax</b>     | BoxEditPrevious (BoxType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                              |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                              |
| <i>BoxType</i>    | Numeric Equivalent or Enumerated Type: Specifies a box type to edit. If a box type is not specified, the macro edits the box immediately previous to the insertion point, regardless of type.<br><br>0 FigureBox!<br>1 TableBox!<br>2 TextBox!<br>3 UserBox!<br>4 EquationBox!<br>5 ButtonBox!<br>6 WatermarkImageBox!<br>7 InlineEquationBox!<br>127 NoBox! |
| <b>Route</b>      | With a box selected, Graphics, Edit Box, select Prev from the Graphics feature bar.                                                                                                                                                                                                                                                                          |
| <b>Example</b>    | To edit the previous occurrence of a button box, the command sequence is:<br><br>BoxEditPrevious (BoxType: ButtonBox!)<br>BoxEnd (State: Save!)                                                                                                                                                                                                              |
| <b>See Also</b>   | ♦ BoxEdit ♦ BoxEditByCounter ♦ BoxEditNext                                                                                                                                                                                                                                                                                                                   |

---

# BoxEnd

---

|                |                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxEnd saves or cancels changes made when creating or editing a graphics box.<br><br>IMPORTANT: This command must be preceded by BoxCreate or BoxEdit. |
| <b>Syntax</b>  | BoxEnd (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; BoxOptionsBar: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )              |

---

## Parameters

|                      |                                                                                                                                                       |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>State</i>         | Numeric Equivalent or Enumerated Type: Specifies whether creating or editing changes to a graphics box are saved or canceled.<br>0 Cancel!<br>1 Save! |
| <i>BoxOptionsBar</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display the box options control bar.<br>0 Hide!<br>1 DontHide!                            |

---

## Route

Graphics, Graphics Styles, select a box type, Edit or Create.

---

## Example

To save changes made to graphics box 1 and hide the Box Options control bar, the command sequence is:

```
BoxEdit (BoxNumber: 1)
BoxEnd (State: Save!; BoxOptionsBar: Hide!)
```

---

## See Also

♦ BoxCreate ♦ BoxEdit

---

# BoxEquationColor

---

## Purpose

BoxEquationColor specifies a box equation color and shade values.

IMPORTANT: This command must be preceded by BoxCreate or BoxEdit, and followed by BoxEnd.

---

## Syntax

BoxEquationColor (ColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; Shade: *Numeric Expression*)

---

## Parameters

|                  |                                                                       |
|------------------|-----------------------------------------------------------------------|
| <i>ColorName</i> | Character Expression: Specifies an equation color.                    |
| <i>Red</i>       | Numeric Expression: Specifies the amount of red in an equation color. |

|              |                                                                         |
|--------------|-------------------------------------------------------------------------|
| <i>Green</i> | Numeric Expression: Specifies the amount of green in an equation color. |
| <i>Blue</i>  | Numeric Expression: Specifies the amount of blue in an equation color.  |
| <i>Shade</i> | Numeric Expression: Specifies the shading of an equation color.         |

---

**Route** Graphics, Equation, Equation Font, Color, select a color.

---

**Example** To specify red as the equation color in box 1, the command sequence is:

```

BoxEdit (BoxNumber: 1)
BoxContentEdit ()
BoxEquationColor (Red: 255; Shade: 100)
BoxEnd (State: Save!)

```

---

**See Also** ♦ BoxContentEdit ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxEquationColorBlue  
♦ ?BoxEquationColorGreen ♦ ?BoxEquationColorRed ♦ ?BoxEquationColorShading

---

## BoxEquationDefaultFont

---

**Purpose** BoxEquationDefaultFont selects a default font and specifies whether to change the default font size.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** BoxEquationDefaultFont (FontSize: *Numeric Expression*)

---

### Parameters

*FontSize* Numeric Expression: Specifies a new default font size. To specify the default font size, select the numeric equivalent or enumerated type.

0 DefaultSize!

---

**Example** To select the default font but specify a non-default 14pt font size, the command is:

```
EquationDefaultFont (FontSize: 14p)
```

---

**See Also** ♦ ?BoxEquationFont

---

## BoxEquationFilename

---

|                   |                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxEquationFilename specifies the name of an equation file.<br><br>IMPORTANT: This command specifies box contents as an equation. To retrieve an equation file, use FileRetrieve. |
| <b>Syntax</b>     | BoxEquationFilename (Filename: <i>Character Expression</i> )                                                                                                                      |
| <b>Parameters</b> |                                                                                                                                                                                   |
| <i>Filename</i>   | Character Expression: Specifies the name of an equation file.                                                                                                                     |
| <b>Route</b>      | With a graphics box selected, choose Graphics, Edit Box, Content, Content, Equation, Filename.                                                                                    |
| <b>Example</b>    | To specify the name of an equation file STAT.DOC, the command is:<br><br>BoxEquationFilename (Filename: "STAT.DOC")                                                               |
| <b>See Also</b>   | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                                    |

---

## BoxEquationFont

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxEquationFont specifies an equation font.<br><br>IMPORTANT: This command is used in conjunction with BoxEquationFontSize.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Syntax</b>     | BoxEquationFont (FontName: <i>Character Expression</i> ; Family: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Attributes: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Weight: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Width: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Source: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; CharacterSet: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>FontName</i>   | Character Expression: Specifies a font name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>Family</i>     | Numeric Equivalent or Enumerated Type: Specifies a font family.<br><br>256 FamilyITCBenquiat!<br>2304 FamilyTimesRoman!<br>4352 FamilyTCGaramond!<br>6400 FamilyCenturySchoolbook!<br>8448 FamilyBodoni!<br>10496 FamilyITCLubalinGraph!<br>12544 FamilyAlbertus!                                                                                                                                                                                                                                                 |



14592 FamilyCourier!  
 16640 FamilyHelvetica!  
 18688 FamilyITCRonda!  
 20736 FamilyMicroStyle!  
 22784 FamilyOptima!  
 24832 FamilyLetterGothic!  
 26880 FamilyBrush!  
 28928 FamilyParkAvenue!  
 30976 FamilySignetRoundhand!  
 33024 FamilyUncial!  
 37120 FamilyOldEnglish!  
 39168 FamilyITCZapfDingbats!  
 41216 FamilyBroadway!  
 43264 FamilyComputer!  
 45312 FamilyPTBarnum!  
 47360 FamilyPonderosa!  
 49408 FamilyMadrone!  
 51456 FamilyHobo!  
 53504 FamilyRevue!

*Attributes*

Numeric Equivalent or Enumerated Type: Specifies font attributes.

0 FontMatchNormal!  
 1 FontMatchItalic!  
 2 FontMatchOutline!  
 4 FontMatchShadow!  
 8 FontMatchSmallCaps!

*Weight*

Numeric Equivalent or Enumerated Type: Specifies a font weight.

0 WeightUnknown!  
 1 WeightUltraThin!  
 8 WeightUltraLight!  
 16 WeightExtraThin!  
 24 WeightThin!  
 32 WeightExtraLight!  
 48 WeightLight!  
 64 WeightDemiLight!  
 80 WeightSemiLight!  
 88 WeightBook!  
 96 WeightRegular!  
 104 WeightRoman!  
 112 WeightMedium!  
 128 WeightDemiBold!  
 136 WeightSemiBold!  
 144 WeightBold!  
 160 WeightExtraBold!  
 176 WeightHeavy!  
 192 WeightExtraHeavy!  
 208 WeightBlack!

224 WeightExtraBlack!  
248 WeightUltraBlack!  
254 WeightMaximum!

*Width*

Numeric Equivalent or Enumerated Type: Specifies a font width.

0 WidthUnknown!  
16 WidthUltraCondensed!  
32 WidthUltraCompressed!  
48 WidthCompressed!  
64 WidthExtraCondensed!  
80 WidthCondensed!  
96 WidthSemiCondensed!  
112 WidthNormal!  
128 WidthSemiExpanded!  
144 WidthExpanded!  
160 WidthExtraExpanded!  
176 WidthUltraExpanded!  
192 WidthDoubleWide!  
224 WidthTripleWide!

*Source*

Numeric Equivalent or Enumerated Type: Specifies whether the font source is a .PRS file or a .DRS file.

16 PRSFile!  
20 DRSFile!

*Type*

Numeric Equivalent or Enumerated Type: Specifies a font type.

134 Speedo!  
135 Intellifont!  
138 Type!  
139 TrueType!

*CharacterSet*

Numeric Equivalent or Enumerated Type: Specifies a character set.

0 FontMatchASCII!  
1 FontMatchMultiNational1!  
2 FontMatchMultiNational2!  
3 FontMatchBoxDrawing!  
4 FontMatchTypographicSymbols!  
5 FontMatchIconicSymbols!  
6 FontMatchMath!  
7 FontMatchMathExtension!  
8 FontMatchGreek!  
9 FontMatchHebrew!  
10 FontMatchCyrillic!  
11 FontMatchJapaneseKana!  
12 FontMatchUserDefined!  
13 FontMatchArabic!  
14 FontMatchArabicScript!

|                 |                                                                                                                                                                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Graphics, Equation, Equation Font, select a font face and a font style from the Equation Font dialog box.                                                                                                                                                                                                                        |
| <b>Example</b>  | To specify an equation font, the command sequence is:<br><br><pre>BoxEquationFont (FontName: "Courier New Bold"; Family: FamilyCourier!; Attributes: FontMatchNormal!; Weight: WeightExtraBold!; Width: WidthUnknown!; Source: DRSFile!; Type: TrueType!; CharacterSet: FontMatchASCII!) BoxEquationFontSize (Size: 18.0p)</pre> |
| <b>See Also</b> | ♦ BoxEquationFontSize                                                                                                                                                                                                                                                                                                            |

## BoxEquationFontSize

|                   |                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxEquationFontSize specifies an equation font size.<br><br><b>IMPORTANT:</b> This command must be preceded by BoxCreate or BoxEdit, and followed by BoxEnd.                           |
| <b>Syntax</b>     | BoxEquationFontSize (Size: <i>Numeric Expression</i> )                                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                        |
| <i>Size</i>       | Numeric Expression: Specifies an equation font size in points.                                                                                                                         |
| <b>Route</b>      | Graphics, Equation, Equation Font, select a font size.                                                                                                                                 |
| <b>Example</b>    | To specify a 12 point equation font in box 1, the command sequence is:<br><br><pre>BoxEdit (BoxNumber: 1) BoxContentEdit () BoxEquationFontSize (Size: 12) BoxEnd (State: Save!)</pre> |
| <b>See Also</b>   | ♦ ?BoxEquationFontSize                                                                                                                                                                 |

## BoxEquationRedisplay

|                |                                                                                                                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxEquationRedisplay redisplay an equation.<br><br><b>IMPORTANT:</b> This command must be preceded by BoxCreate or BoxEdit, and followed by BoxEnd.<br><br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                   |                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | BoxEquationRedisplay (Zoom: <i>Numeric Expression</i> )                                                                                                                                  |
| <b>Parameters</b> |                                                                                                                                                                                          |
| <i>Zoom</i>       | Numeric Expression: Specifies a degree of enlargement or reduction of the equation.                                                                                                      |
| <b>Example</b>    | To redisplay an equation in box 1 and reduce it by 50%, the command is:<br><br><pre>BoxEdit (BoxNumber: 1) BoxContentEdit () BoxEquationRedisplay (Zoom: 50) BoxEnd (State: Save!)</pre> |
| <b>See Also</b>   | ♦ BoxContentEdit ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                          |

## BoxEquationSaveAsGraphic

|                   |                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxEquationSaveAsGraphic saves an equation in a graphics file format.                                                                                                                                                                               |
| <b>Syntax</b>     | BoxEquationSaveAsGraphic (Filename: <i>Character Expression</i> ; Format: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                                                                                     |
| <i>Filename</i>   | Character Expression: Names a file to save.                                                                                                                                                                                                         |
| <i>Format</i>     | Numeric Equivalent or Enumerated Type: Specifies the graphic format of an equation box. If a format is not specified, the equation is saved in the current graphics format.<br><br><pre>725 WordPerfectGraphic_10! 726 WordPerfectGraphic_20!</pre> |
| <b>Route</b>      | Graphics, Equation, File, Save As, enter filename, select graphics format.                                                                                                                                                                          |
| <b>Example</b>    | To save an equation named Calculus as a WordPerfect Graphic 1.0, the command is:<br><br><pre>BoxEquationSaveAsGraphic (Filename: "Calculus"; Format: WordPerfectGraphic_10!)</pre>                                                                  |
| <b>See Also</b>   | ♦ BoxEquationFilename                                                                                                                                                                                                                               |

---

# BoxFill

---

|                      |                                                                                                                                                                                     |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>       | BoxFill specifies a box fill style.<br><br>IMPORTANT: This command must be preceded by BoxCreate or BoxEdit, and followed by BoxEnd.                                                |
| <b>Syntax</b>        | BoxFill (FillStyleName: <i>Character Expression</i> )                                                                                                                               |
| <b>Parameters</b>    |                                                                                                                                                                                     |
| <i>FillStyleName</i> | Character Expression: Specifies the name of a box fill style.                                                                                                                       |
| <b>Route</b>         | Graphics, choose a box type, Border/Fill, select a fill style.                                                                                                                      |
| <b>Example</b>       | To specify the Waves fill style for box 1, the command sequence is:<br><br>BoxEdit (BoxNumber: 1)<br>BoxContentEdit ()<br>BoxFill (FillStyleName: "Waves")<br>BoxEnd (State: Save!) |
| <b>See Also</b>      | ♦ BoxContentEdit ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                     |

---

# BoxHeight

---

|                   |                                                                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxHeight sets box height based on the height of box contents or manually specified measurements.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.                      |
| <b>Syntax</b>     | BoxHeight (Height: <i>Numeric Equivalent or Enumerated Type or Numeric Expression</i> )                                                                                                                                                                                         |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                 |
| <i>Height</i>     | Numeric Equivalent or Enumerated Type or Numeric Expression: Specifies a graphics box height. Select a numeric equivalent or enumerated type to set box height based on the height of its contents. Enter a numeric expression to manually set the height.<br><br>2 AutoHeight! |
| <b>Route</b>      | Graphics, choose a box type, Size, select a height option. If you select Set, enter a box height.                                                                                                                                                                               |

---

**Example**

To set box height to four inches, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxHeight (Height: 4")
BoxEnd (State: Save!)
```

---

**See Also**

♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxHeight

---

## BoxHorizontalAlignment

---

**Purpose**

BoxHorizontalAlignment specifies the alignment of a graphics box attached to a page or paragraph.

**IMPORTANT:** This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax**

BoxHorizontalAlignment (Alignment: *Numeric Equivalent* or *Enumerated Type*, Position: *Numeric Equivalent* or *Enumerated Type*, Offset: *Numeric Expression*; Left Column: *Numeric Expression*; Right Column: *Numeric Expression*)

---

**Parameters***Alignment*

Numeric Equivalent or Enumerated Type: Specifies where to horizontally align a graphics box.

- 1 AlignMargins!
- 2 AlignColumns!

*Position*

Numeric Equivalent or Enumerated Type: Specifies the horizontal position of a graphics box after its alignment to margins or columns.

- 0 Left!
- 2 Center!
- 3 Right!
- 7 FullAlign!

*Offset*

Numeric Expression: Specifies the left or right offset of a graphics box. The offset moves the image to the left or right of the alignment point. Left offsets are negative numbers, right offsets are positive.

*Left Column*

Numeric Expression: Specifies the left column when a box is aligned to columns.

*Right Column*

Numeric Expression: Specifies the right column when a box is aligned to columns.

---

**Route**

Graphics, choose a box type, Position, select horizontal position options. Position options are not available for character-anchored graphic boxes.

---

|                |                                                                                                                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To center a graphics box between two columns, the command sequence is:<br><br>BoxCreate (BoxType: FigureBox!)<br>BoxHorizontalAlignment (Alignment: AlignColumns!; Position: Center!; Offset: 0"; LeftColumn: 1;<br>RightColumn: 2)<br>BoxEnd (State: Save!) |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxHorizontalPosition<br>♦ ?BoxHorizontalAlignment ♦ ?BoxHorizontalAlignTo ♦ ?BoxHorizontalOffset |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|

---

## BoxHorizontalPosition

---

|                |                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxHorizontalPosition specifies the horizontal position of a graphics box relative to the left edge of the page. |
|----------------|------------------------------------------------------------------------------------------------------------------|

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

|               |                                                                  |
|---------------|------------------------------------------------------------------|
| <b>Syntax</b> | BoxHorizontalPosition (Position: <i>Measurement Expression</i> ) |
|---------------|------------------------------------------------------------------|

---

|                   |  |
|-------------------|--|
| <b>Parameters</b> |  |
|-------------------|--|

|                 |                                                                              |
|-----------------|------------------------------------------------------------------------------|
| <i>Position</i> | Measurement Expression: Specifies the horizontal position of a graphics box. |
|-----------------|------------------------------------------------------------------------------|

---

|              |                                                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Graphics, select a box type, Position, Horizontal, specify position, select Left Edge of Page from the From pop-up list. |
|--------------|--------------------------------------------------------------------------------------------------------------------------|

---

|                |                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To position a graphics box two inches from the left edge of the page, the command sequence is:<br><br>BoxCreate (BoxType: FigureBox!)<br>BoxHorizontalPosition (Position: 2.5")<br>BoxEnd (State: Save!) |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxHorizontalAlignment |
|-----------------|---------------------------------------------------------|

---

## BoxImageBlackWhiteThreshold

---

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxImageBlackWhiteThreshold determines the threshold or point at which colors and/or grays in the current graphics image are converted to black and white. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

|                   |                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | BoxImageBlackWhiteThreshold (Threshold: <i>Numeric Expression</i> )                                                                                                                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                     |
| <i>Threshold</i>  | Numeric Expression: Specifies the threshold at which colors and/or grays are converted to either black or white. Colors or grays below the threshold are converted to black. Those above the threshold are converted to white. Values range from 1 to 255, with 127 as the default. |
| <b>Route</b>      | Graphics, Figure, select a filename, OK, Tools, Black & White, select a threshold.                                                                                                                                                                                                  |
| <b>Example</b>    | To set the threshold for converting colors and/or grays to black and white to 150, the command sequence is:<br><br><pre>BoxCreate (BoxType: FigureBox!) BoxImageBlackWhiteThreshold (Threshold: 150) BoxEnd (State: Save!)</pre>                                                    |
| <b>See Also</b>   | <ul style="list-style-type: none"> <li>♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxImageMonochrome</li> <li>♦ ?BoxImageBlackWhiteThreshold</li> </ul>                                                                                                                                       |

---

## BoxImageBrightness

|                   |                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxImageBrightness adjusts the brightness (saturation) of a color or black and white image.<br><br><b>IMPORTANT:</b> This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
| <b>Syntax</b>     | BoxImageBrightness (Amount: <i>Numeric Expression</i> )                                                                                                                                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                                                                                             |
| <i>Amount</i>     | Numeric Expression: Specifies the brightness of an image color. Values range from -1.0 (all black) to 1.0 (all white).                                                                                                                                      |
| <b>Route</b>      | Graphics, Edit Box, Tools, Brightness.                                                                                                                                                                                                                      |
| <b>Example</b>    | To set the brightness of an image color to .5, the command sequence is:<br><br><pre>BoxEdit (BoxNumber: 1) BoxImageBrightness (Amount: 0.5) BoxEnd (State: Save!)</pre>                                                                                     |



---

**See Also**

♦ `BoxImageContrast` ♦ `BoxCreate` ♦ `BoxEdit` ♦ `BoxEnd` ♦ `?BoxImageBrightness`

---

## BoxImageContrast

---

**Purpose** `BoxImageContrast` determines the contrast between light and dark areas of a graphic image.

**IMPORTANT:** This command must be preceded by a command which specifies a box, such as `BoxCreate`, and followed by a closing command such as `BoxEnd`.

---

**Syntax** `BoxImageContrast (Contrast: Numeric Expression)`

---

**Parameters**

*Contrast* Numeric Expression: Specifies the contrast or sharpness of an image. Values range from -1.0 (slight contrast) to 1.0 (substantial contrast).

---

**Route** Graphics, Edit Box, Tools, Contrast.

---

**Example** To set the contrast of an image color to 0.7, the command sequence is:

```
BoxEdit (BoxNumber: 1)
BoxImageContrast (Contrast: 0.7)
BoxEnd (State: Save!)
```

---

**See Also**

♦ `BoxCreate` ♦ `BoxEdit` ♦ `BoxEnd` ♦ `?BoxImageContrast`

---

## BoxImageFill

---

**Purpose** `BoxImageFill` specifies a graphic image fill type.

---

**Syntax** `BoxImageFill (ImageFillType: Numeric Equivalent or Enumerated Type)`

---

**Parameters**

*ImageFillType* Numeric Equivalent or Enumerated Type: Specifies an image fill type.

- 0 Normal!
- 1 Transparent!
- 2 White!

---

|                 |                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Graphics, Edit Box, Tools, Fill.                                                                                                                                        |
| <b>Example</b>  | To set an image fill type to transparent, the command sequence is:<br><br>BoxEdit (BoxNumber: 1)<br>BoxImageFill (ImageFillType: Transparent!)<br>BoxEnd (State: Save!) |
| <b>See Also</b> | ♦ ?BoxImageFill                                                                                                                                                         |

---

## BoxImageFlipX

---

|                |                                                                                                                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxImageFlipX specifies whether an image is reversed on its X axis. For example, if an image faces right, this command causes it to face left.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                                                                             |
|---------------|-----------------------------------------------------------------------------|
| <b>Syntax</b> | BoxImageFlipX (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|-----------------------------------------------------------------------------|

---

### Parameters

|              |                                                                                                                              |
|--------------|------------------------------------------------------------------------------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether an image is reversed on its horizontal axis.<br><br>0 No!<br>1 Yes! |
|--------------|------------------------------------------------------------------------------------------------------------------------------|

---

|              |                                             |
|--------------|---------------------------------------------|
| <b>Route</b> | Graphics, Edit Box, Tools, Mirror Vertical. |
|--------------|---------------------------------------------|

---

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To reverse an image on its horizontal axis, the command sequence is:<br><br>BoxEdit (BoxNumber: 1)<br>BoxImageFlipX (State: Yes!)<br>BoxEnd (State: Save!) |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                                 |
|-----------------|-------------------------------------------------|
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxImageFilpX |
|-----------------|-------------------------------------------------|

---

---

## BoxImageFlipY

---

**Purpose** BoxImageFlipY specifies whether to reverse an image on its Y axis, which turns the image upside down.

IMPORTANT: This command must be preceded by BoxCreate or BoxEdit, and followed by BoxEnd.

---

**Syntax** BoxImageFlipY (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type:

0 No!  
1 Yes!

---

**Route** Graphics, Edit Box, Tools, Mirror Horizontal.

---

**Example** To reverse a image along its Y axis, the command is:

```
BoxEdit (BoxNumber: 1)
BoxImageFlipY (State: Yes!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxImageFlipX ♦ ?BoxImageFlipY

---

---

## BoxImageInvertColors

---

**Purpose** BoxImageInvertColors specifies whether the current colors of an image are inverted to their respective complementary colors.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxImageInvertColors (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type: Specifies whether image colors are inverted.

0 No!  
1 Yes!

|                 |                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Graphics, Edit Box, Tools, Complement.                                                                                                                            |
| <b>Example</b>  | To invert the colors of the current image, the command sequence is:<br><pre>BoxEdit (BoxNumber: 1) BoxImageInvertColors (State: Yes!) BoxEnd (State: Save!)</pre> |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxImageInvertColors                                                                                                            |

---

## BoxImageMonochrome

|                   |                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxImageMonochrome converts a color image to black and white.<br><p>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.</p> |
| <b>Syntax</b>     | BoxImageMonochrome (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                          |
| <b>Parameters</b> |                                                                                                                                                                                                                           |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether a color image is converted to black and white.<br><pre>0 No! 1 Yes!</pre>                                                                                        |
| <b>Route</b>      | Graphics, Edit Box, Tools, Black & White, select a threshold.                                                                                                                                                             |
| <b>Example</b>    | To convert a color image to black and white, the command sequence is:<br><pre>BoxEdit (BoxNumber: 1) BoxImageMonochrome (State: Yes!) BoxEnd (State: Save!)</pre>                                                         |
| <b>See Also</b>   | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxImageBlackWhiteThreshold<br>♦ BoxImageFill ♦ ?BoxImageMonochrome                                                                                                                      |

---

## BoxImageMove

|                |                                                                                                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxImageMove moves an image within a graphics box.<br><p>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.</p> |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                   |                                                                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | BoxImageMove (XPosition: <i>Numeric Expression</i> ; YPosition: <i>Numeric Expression</i> )                                                                                                               |
| <b>Parameters</b> |                                                                                                                                                                                                           |
| <i>XPosition</i>  | Numeric Expression: Specifies the horizontal position of an image within a box. Values range from -5.0 to 5.0, with 0.0 being the default.                                                                |
| <i>YPosition</i>  | Numeric Expression: Specifies the vertical position of an image within a box. Values range from -5.0 to 5.0, with 0.0 centering the image.                                                                |
| <b>Route</b>      | Graphics, Edit Box, Tools, Move.                                                                                                                                                                          |
| <b>Example</b>    | To move an image slightly up and to the left within a graphics box, the command sequence is:<br><br><pre>BoxEdit (BoxNumber: 2) BoxImageMove (XPosition: 0.2; YPosition: 0.2) BoxEnd (State: Save!)</pre> |
| <b>See Also</b>   | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxImageTranslationX<br>♦ ?BoxImageTranslationY                                                                                                                         |

---

## BoxImageRetrieve

|                   |                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxImageRetrieve retrieves a specified image from a specific location.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
| <b>Syntax</b>     | BoxImageRetrieve (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Filename: <i>Character Expression</i> )                                                                                                         |
| <b>Parameters</b> |                                                                                                                                                                                                                                 |
| <i>Action</i>     | Numeric Equivalent or Enumerated Type: Specifies the current location of an image.<br><br><pre>0 LeaveExternal! 1 MakeInternal! 2 UseInternal!</pre>                                                                            |
| <i>Filename</i>   | Character Expression: Specifies the filename of an image to retrieve.                                                                                                                                                           |
| <b>Route</b>      | Graphics, Edit Box, Content, Content, Image, enter a filename.                                                                                                                                                                  |
| <b>Example</b>    | To retrieve an image named MOUNTAIN.WPG from the current document into another box created in the same document, the command sequence is:                                                                                       |

BoxCreate (BoxType: FigureBox!)  
BoxImageRetrieve (Action: UseInternal!; Filename: "MOUNTAIN.WPG")  
BoxEnd (State: Save!)

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd

---

## BoxImageRotate

---

**Purpose** BoxImageRotate rotates a graphic image.  
IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxImageRotate (Rotation: *Numeric Expression*)

---

### Parameters

*Rotation* Numeric Expression: Specifies a degree of rotation. Values range from 0.0 to 359.0.

---

**Route** Graphics, Edit Box, Tools, Rotate.

---

**Example** To rotate an image 90 degrees, the command sequence is:

```
BoxEdit (BoxNumber: 1)
BoxImageRotate (Rotation: 90.0)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxImageRotation

---

## BoxImageSave

---

**Purpose** BoxImageSave saves an image either with its current name or with a new name. It can also specify the graphics format.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxImageSave (Filename: *Character Expression*)

---

## Parameters

*Filename* Character Expression: Names an image.

---

**Example** To save an image named FIREWORKS.WPG, the command sequence is:

```
BoxEdit (BoxNumber: 3)
BoxImageSave (Filename: "FIREWORKS.WPG")
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd

---

---

## BoxImageScaling

**Purpose** BoxImageScaling scales an image within a graphics box.  
IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax** BoxImageScaling (XScale: *Numeric Expression*; YScale: *Numeric Expression*)

---

## Parameters

*XScale* Numeric Expression: Specifies the width to which an image is scaled. Values range from 0.01 to 9.99, with 1.0 being the default.

*YScale* Numeric Expression: Specifies the height to which an image is scaled. Values range from 0.01 to 9.99, with 1.0 being the default.

---

**Route** Graphics, Edit Box, Tools, Scale.

---

**Example** To scale an image to a width of 1.5 and a height of 1.2, the command is:

```
BoxEdit (BoxNumber: 2)
BoxImageScaling (XScale: 1.5; YScale: 1.2)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxImageScalingX ♦ ?BoxImageScalingY

---

---

## BoxImageShowBackground

**Purpose** BoxImageShowBackground specifies whether any background colors or gradients saved with the original image are visible.

IMPORTANT: This command must be preceded by a command which specifies a box, such as `BoxCreate`, and followed by a closing command such as `BoxEnd`.

---

|                   |                                                                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | <code>BoxImageShowBackground</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                           |
| <b>Parameters</b> |                                                                                                                                                                                             |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether image backgrounds are visible.<br><br>0 No!<br>1 Yes!                                                                              |
| <b>Route</b>      | Graphics, Edit Box, Tools, Image Settings, Miscellaneous, Use Page Background.                                                                                                              |
| <b>Example</b>    | To show an image background, the command sequence is:<br><br><code>BoxEdit (BoxNumber: 1)</code><br><code>BoxImageShowBackground (State: Yes!)</code><br><code>BoxEnd (State: Save!)</code> |
| <b>See Also</b>   | ♦ <code>BoxCreate</code> ♦ <code>BoxEdit</code> ♦ <code>BoxEnd</code> ♦ <code>?BoxImagePageBackground</code>                                                                                |

---

## BoxInitialStyleDlg

---

|                |                                                                                                                                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>BoxInitialStyleDlg</code> displays the Text Box Initial Style dialog box.<br><br>IMPORTANT: A graphics box must be selected before this command can be executed.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | <code>BoxInitialStyleDlg ()</code>                                                                                                                                                                                                                                             |

---

## BoxOLEEditObject

---

|                |                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>BoxOLEEditObject</code> uses the OLE (Object Linking and Embedding) server to edit graphics box contents. |
| <b>Syntax</b>  | <code>BoxOLEEditObject ()</code>                                                                                |

---



---

|              |                                                                                     |
|--------------|-------------------------------------------------------------------------------------|
| <b>Route</b> | When a box contains an OLE object, choose Graphics, Edit Box, Content button, Edit. |
|--------------|-------------------------------------------------------------------------------------|

---

## BoxOverlap

---

|                |                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxOverlap specifies whether a graphics box is allowed to overlap other boxes.<br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                                                                          |
|---------------|--------------------------------------------------------------------------|
| <b>Syntax</b> | BoxOverlap (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|--------------------------------------------------------------------------|

---

### Parameters

|              |                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether the box is allowed to overlap other boxes.<br>0 No!<br>1 Yes! |
|--------------|------------------------------------------------------------------------------------------------------------------------|

---

|              |                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Graphics, Edit Box, Position, Page Anchor or Paragraph Anchor, select Allow Box to Overlap Other Boxes. |
|--------------|---------------------------------------------------------------------------------------------------------|

---

|                |                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To allow a graphics box to overlap other boxes, the command sequence is:<br><br>BoxCreate (BoxType: FigureBox!)<br>BoxOverlap (State: Yes!)<br>BoxEnd (State: Save!) |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------|
| <b>See Also</b> | ♦ BoxHorizontalAlignment ♦ BoxHorizontalPosition ♦ BoxVerticalAlignment<br>♦ BoxVerticalPosition ♦ ?BoxOverlap |
|-----------------|----------------------------------------------------------------------------------------------------------------|

---

## BoxPosDlg

---

|                |                                                 |
|----------------|-------------------------------------------------|
| <b>Purpose</b> | BoxPosDlg displays the Box Position dialog box. |
|----------------|-------------------------------------------------|

---

|               |              |
|---------------|--------------|
| <b>Syntax</b> | BoxPosDlg () |
|---------------|--------------|

---

|              |                                        |
|--------------|----------------------------------------|
| <b>Route</b> | Graphics, select a box type, Position. |
|--------------|----------------------------------------|

---

---

## BoxResetBorder

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | <p>BoxResetBorder resets the border values for the current box to the border values for the corresponding box style. For example, border changes made to a specific figure box reset to the default border values assigned to the Figure Box style.</p> <p>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.</p> <p>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.</p> |
| <b>Syntax</b>   | BoxResetBorder ()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

---

## BoxResetCaption

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | <p>BoxResetCaption resets the caption values for the current box to the caption values for the corresponding box style. For example, caption changes made to a specific text box reset to the default caption values assigned to the Text Box style.</p> <p>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.</p> |
| <b>Syntax</b>   | BoxResetCaption ()                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Route</b>    | Graphics, Edit Box, Caption, Reset.                                                                                                                                                                                                                                                                                                                                                                               |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                                                                                                                                                                                                                                                                    |

---

## BoxResetContent

---

|                |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <p>BoxResetContent resets the content values for the current box to the content values for the corresponding box style. For example, content changes made to a specific user box reset to the default content values assigned to the User Box style.</p> <p>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.</p> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                     |
|-----------------|-------------------------------------|
| <b>Syntax</b>   | BoxResetContent ()                  |
| <b>Route</b>    | Graphics, Edit Box, Content, Reset. |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd      |

---

## BoxResetContentRender

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | BoxResetContentRender resets the content render values for the current box to the content render values for the corresponding box style. For example, content render changes made to a specific equation box reset to the default content render values assigned to the Equation Box style.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
| <b>Syntax</b>   | BoxResetContentRender ()                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Route</b>    | After making border changes to a box, choose Graphics.                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                                                                                                                                                                                                                                                                                                       |

---

## BoxResetCounter

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | BoxResetCounter resets the counter values for the current box to the counter values for the corresponding box style. For example, counter changes made to a specific equation box reset to the default counter values assigned to the Equation Box style.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
| <b>Syntax</b>   | BoxResetCounter ()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>See Also</b> | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

---

---

## BoxResetFill

---

**Purpose**

BoxResetFill resets the fill values for the current box to the position values for the corresponding box style. For example, fill style changes made to a specified figure box reset to the default fill values assigned to the Figure Box style.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax**

BoxResetFill ()

---

**See Also**

♦ BoxCreate ♦ BoxEdit ♦ BoxEnd

---

---

## BoxResetPosition

---

**Purpose**

BoxResetPosition resets the position values for the current box to the position values for the corresponding box style. For example, position changes made to a specific figure box reset to the default position values assigned to the Figure Box style.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax**

BoxResetPosition ()

---

**See Also**

♦ BoxCreate ♦ BoxEdit ♦ BoxEnd

---

---

## BoxResetTextFlow

---

**Purpose**

BoxResetTextFlow resets the text flow values for the current box to the text flow values for the corresponding box style. For example, text flow changes made to a specific user box reset to the default text flow values assigned to the User Box style.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

---

**Syntax**                      BoxResetTextFlow ()

---

**See Also**                    ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd

---

---

## BoxStyleCopy

---

**Purpose**                      BoxStyleCopy copies the specified style from one location to another.

---

**Syntax**                      BoxStyleCopy (Style: *Character Expression*; FromLibrary: *Numeric Equivalent or Enumerated Type*; ToLibrary: *Numeric Equivalent or Enumerated Type*; DestName: *Character Expression*)

---

### Parameters

*Style*                              Character Expression: Specifies the name of the style to copy.

*FromLibrary*                    Numeric Equivalent or Enumerated Type: Specifies the current location of the style to copy.

- 0 CurrentDocument!
- 1 PersonalLibrary!
- 2 SharedLibrary!

*ToLibrary*                      Numeric Equivalent or Enumerated Type: Specifies the destination of the style to copy.

- 0 CurrentDocument!
- 1 PersonalLibrary!
- 2 SharedLibrary!

*DestName*                      Character Expression: Specifies the new name to give to the style. This parameter is only necessary if you want to place a second copy of the style in the same location. The macro expects this parameter if the contents of the FromLibrary and ToLibrary parameters are the same.

---

**Route**                        Graphics, Graphics Styles, Box, select a style, Options, Copy, enter a new style name.

---

**Example**                      To copy a style named Original to a style named New, the command is:

```
BoxStyleCopy (Style: "Original"; FromLibrary: PersonalLibrary!; ToLibrary: PersonalLibrary!;
DestName:"New")
```

---

**See Also**                    ♦ BoxStyleCreate ♦ BoxStyleDelete ♦ BoxStyleEdit ♦ BoxStyleDlg  
♦ BoxStyleRetrieve

---

---

# BoxStyleCreate

|                   |                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxStyleCreate specifies the name of the style to create.                                                                                       |
| <b>Syntax</b>     | BoxStyleCreate (Style: <i>Character Expression</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                             |
| <b>Parameters</b> |                                                                                                                                                 |
| <i>Style</i>      | Character Expression: Specifies the name of the style to create.                                                                                |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies a library to contain the new style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary! |
| <b>Route</b>      | Graphics, Graphics Styles, Box, Create, select options.                                                                                         |
| <b>Example</b>    | To create a box style named NewBox, the command is:<br>BoxStyleCreate (Style: "NewBox"; Library: PersonalLibrary!)                              |
| <b>See Also</b>   | ♦ BoxStyleDelete ♦ GraphicBoxStyleDlg ♦ BoxStyleEdit ♦ BoxStyleRetrieve                                                                         |

---

# BoxStyleDelete

|                   |                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxStyleDelete deletes a specified box style.                                                                                                         |
| <b>Syntax</b>     | BoxStyleDelete (Style: <i>Character Expression</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                   |
| <b>Parameters</b> |                                                                                                                                                       |
| <i>Style</i>      | Character Expression: Specifies the name of the style to delete.                                                                                      |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies a library from which to delete the style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary! |
| <b>Route</b>      | Graphics, Graphics Styles, Box, select a style, Options, Delete.                                                                                      |

---

**Example** To delete a box style named OldBox, the command is:

BoxStyleDelete (Style: "OldBox"; Library: CurrentDoc!)

---

**See Also** ♦ BoxStyleCreate ♦ GraphicBoxStyleDlg ♦ BoxStyleEdit ♦ BoxStyleRetrieve

---

## BoxStyleEdit

---

**Purpose** BoxStyleEdit initiates the editing of a specified style.

IMPORTANT: This command must be followed by style editing commands.

---

**Syntax** BoxStyleEdit (Style: *Numeric Equivalent* or *Enumerated Type* or *Character Expression*;  
Library: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies the name of the style to edit. Select a numeric equivalent or enumerated type to specify styles provided by WordPerfect. To edit a user-defined style, enclose the name in quotation marks.

- 0 FigureBox!
- 1 TableBox!
- 2 TextBox!
- 3 UserBox!
- 4 EquationBox!
- 5 ButtonBox!
- 6 WatermarkImageBox!
- 7 InlineEquationBox!
- 127 NoBox!

*Library* Numeric Equivalent or Enumerated Type: Specifies the library containing the style to edit.

- 0 CurrentDoc!
  - 1 PersonalLibrary!
  - 2 SharedLibrary!
- 

**Route** Graphics, Graphics Styles, Box, select a style, Edit, select options.

---

**Example** To edit a box style named Watermark, located in the Personal Library, the command is:

BoxStyleEdit (Style: "Watermark"; Library: PersonalLibrary!)

---

---

**See Also**

- ♦ `BoxStyleCreate` ♦ `BoxStyleDelete` ♦ `BoxStyleEdit` ♦ `BoxStyleEnd`
- ♦ `BoxStyleRetrieve`

---

## BoxStyleEnd

---

**Purpose**

`BoxStyleEnd` specifies whether to save changes made to a box style.

IMPORTANT: This command must be preceded by `BoxStyleCreate` or `BoxStyleEdit`, and by style editing commands.

---

**Syntax**

`BoxStyleEnd` (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type:

- 0 Cancel!
- 1 Save!

---

**Route**

Graphics, Graphics Styles, Box, select a style, Create or Edit, specify editing options.

---

**Example**

To save changes made to a box style named `Watermark`, the command sequence is:

```
BoxStyleEdit (Style: "Watermark"; Library: PersonalLibrary!)
BoxBorder (Style: 5)
BoxStyleEnd (State: Save!)
```

---

**See Also**

- ♦ `BoxStyleCreate` ♦ `BoxStyleEdit`

---

## BoxStyleName

---

**Purpose**

`BoxStyleName` names a style.

IMPORTANT: This command must be preceded by `BoxStyleCreate`.

---

**Syntax**

`BoxStyleName` (StyleName: *Character Expression*)

---

**Parameters***StyleName*

Character Expression: Specifies a new name for a style.



|                 |                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Graphics, Graphics Styles, Box, Create, enter a style name.                                                                                                                |
| <b>Example</b>  | To create a style named Portrait, the command sequence is:<br><br><pre>BoxStyleCreate (Style: "unnamed"; Library: PersonalLibrary!) BoxStyleName (Style: "Portrait")</pre> |
| <b>See Also</b> | ♦ BoxStyleCreate                                                                                                                                                           |

## BoxStyleRetrieve

|                   |                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxStyleRetrieve retrieves the specified style into the current document or your personal or shared style libraries.                                                                  |
| <b>Syntax</b>     | BoxStyleRetrieve (Filename: <i>Character Expression</i> ; Library: <i>Numeric Equivalent or Enumerated Type</i> )                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                       |
| <i>Filename</i>   | Character Expression: Specifies the style library containing the styles to retrieve.                                                                                                  |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies the location into which the style should be retrieved.<br><br><pre>0 CurrentDoc! 1 PersonalLibrary! 2 SharedLibrary!</pre>           |
| <b>Route</b>      | Graphics, Graphics Styles, Options, Retrieve, type a filename.                                                                                                                        |
| <b>Example</b>    | To retrieve all styles in a style library named Retro into your Shared Style Library, the command is:<br><br><pre>BoxStyleRetrieve (Filename: "Retro"; Library: SharedLibrary!)</pre> |
| <b>See Also</b>   | ♦ BoxStyleCopy ♦ BoxStyleSave                                                                                                                                                         |

## BoxStyleSave

|                |                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | BoxStyleSave saves all styles from the current document or your personal or shared style libraries to the specified style library.<br><br><b>IMPORTANT:</b> If the specified style library contains styles, they will be replaced. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Syntax**                      `BoxStyleSave` (Filename: *Character Expression*; Library: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Filename*                      Character Expression: Specifies the style library to which the styles are saved.

*Library*                      Numeric Equivalent or Enumerated Type: Specifies the current location of the styles.

- 0    `CurrentDoc!`
- 1    `PersonalLibrary!`
- 2    `SharedLibrary!`

---

**Route**                      Graphics, Graphics Styles, Options, Save As, type a filename.

---

**Example**                      To save all in your Shared Style library to a style library named Award, the command is:

`BoxStyleSave` (Filename: "Award"; Library: SharedLibrary!)

---

**See Also**                      ♦ `BoxStyleCopy` ♦ `BoxStyleRetrieve`

---

---

## BoxTextAngle

---

**Purpose**                      `BoxTextAngle` rotates the text in a graphics box.

IMPORTANT: This command must be preceded by `BoxStyleCreate` or `BoxStyleEdit` and followed by `BoxEnd`.

---

**Syntax**                      `BoxTextAngle` (Angle: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Angle*                      Numeric Equivalent or Enumerated Type: Specifies the degree to rotate the text in a graphics box.

- 0    `None!`
- 1    `Degrees90!`
- 2    `Degrees180!`
- 3    `Degrees270!`

---

**Route**                      Graphics, Edit Box, Content, select an option from Rotate Contents Counterclockwise.

---

---

**Example**

To rotate box text 180 degrees, the command sequence is:

```
BoxCreate (BoxType: TextBox!)
BoxTextAngle (Angle: Degrees180!)
BoxEnd (State: Save!)
```

---

**See Also**

♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxTextAngle

---

## BoxTextDefaultStyle

---

**Purpose**

BoxTextDefaultStyle determines the default style for box text.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

---

**Syntax**

BoxTextDefaultStyle (Style: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Style*

Numeric Equivalent or Enumerated Type: Specifies the default style for box text.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!

41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutLnStyle!  
46 OutlineStyle!  
47 LegalOutLnStyle!  
48 BulletsOutLnStyle!  
49 HeadingsOutLnStyle!  
50 Legal\_2OutLnStyle!  
51 NumbersOutLnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

---

**Route**

Graphics, Graphics Styles, Box, select a style, Edit or Create a text box, Settings, Change, select a style, Apply.

---

**Example**

To specify Normal Style as the default for box text, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxTextDefaultStyle (Style: NormalStyle!)
BoxEnd (State: Save!)
```

---

**See Also**

♦ `BoxCreate` ♦ `BoxEdit` ♦ `BoxEnd` ♦ `?BoxTextDefaultStyle`

---

## BoxTextEdit

---

**Purpose** `BoxTextEdit` opens a text box.

IMPORTANT: This command must be followed by commands which modify box contents, such as `Type`, and by `BoxTextEditEnd`.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** `BoxTextEdit ()`

---

**See Also** ♦ `BoxTextEditEnd` ♦ `BoxContentEdit`

---

## BoxTextEditEnd

---

**Purpose** `BoxTextEditEnd` closes a text box.

IMPORTANT: This command must be preceded by `BoxTextEdit` and commands such as `Type` which modify box text.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** `BoxTextEditEnd ()`

---

**See Also** ♦ `BoxTextEdit` ♦ `SubstructureExit`

---

## BoxTextFilename

---

**Purpose** `BoxTextFilename` specifies the name of a text file. To retrieve a text file, use `FileRetrieve`.

IMPORTANT: This command must be preceded by a command which specifies a box, such as `BoxCreate`, and followed by a closing command such as `BoxEnd`.

---

|                   |                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | BoxTextFilename (Filename: <i>Character Expression</i> )                                                                                                                        |
| <b>Parameters</b> |                                                                                                                                                                                 |
| <i>Filename</i>   | Character Expression: Specifies the name of the text file.                                                                                                                      |
| <b>Route</b>      | Graphics, select a box type, Contents, specify the box contents as Text, type a filename in the Filename text box.                                                              |
| <b>Example</b>    | To specify a text filename of DRAFT.FIL, the command sequence is:<br><br><pre>BoxCreate (BoxType: TextBox!) BoxTextFilename (Filename: "DRAFT.FIL") BoxEnd (State: Save!)</pre> |

---

## BoxTextFlow

|                   |                                                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxTextFlow specifies the flow or placement of text surrounding a graphics box.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd. |
| <b>Syntax</b>     | BoxTextFlow (Wrap: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                                                                          |
| <i>Wrap</i>       | Numeric Equivalent or Enumerated Type: Specifies the location of the text flow surrounding a graphics box.<br><br><pre>0 LargestSide! 1 LeftSide! 2 RightSide! 3 NeitherSide! 8 BothSides! 9 Columnar! 15 NoWrapping!</pre>              |
| <b>Route</b>      | Graphics, choose a box type, Wrap, select a side to wrap text around.                                                                                                                                                                    |
| <b>Example</b>    | To allow text to wrap around the box, the command sequence is:<br><br><pre>BoxCreate (BoxType: FigureBox!) BoxTextFlow (Wrap: BothSides!) BoxEnd (State: Save!)</pre>                                                                    |

---

**See Also** ♦ [BoxCreate](#) ♦ [BoxEdit](#) ♦ [BoxEnd](#) ♦ [?BoxWrapping](#)

---

## BoxTextFlowContour

---

**Purpose** `BoxTextFlowContour` specifies whether text should be contoured to the shape of the graphics image. Contouring applies only to text outside of the box.

---

**Syntax** `BoxTextFlowContour ()`

---

**Route** Graphics, choose a box type, Wrap, Contour.

---

**See Also** ♦ [BoxTextFlow](#) ♦ [BoxTextFlowSquare](#)

---

## BoxTextFlowSquare

---

**Purpose** `BoxTextFlowSquare` specifies whether text should be contoured to the square shape of the box.

IMPORTANT: `BoxTextFlowSquare` is only necessary if you have previously used `BoxTextFlowContour`.

---

**Syntax** `BoxTextFlowSquare ()`

---

**Route** Graphics, choose a box type, Wrap, Square.

---

**See Also** ♦ [BoxTextFlow](#) ♦ [BoxTextFlowContour](#)

---

## BoxUpdateDisplay

---

**Purpose** `BoxUpdateDisplay` shows the most current version of a box.

---

**Syntax** `BoxUpdateDisplay ()`

---

**Route** `BoxUpdateDisplay` records when a graphics box is updated, so the route will vary. For example, the following is one of many valid routes: Graphics, Edit Box, Size, specify box width or height.

---

# BoxVerticalAlignment

**Purpose** BoxVerticalAlignment aligns a graphics box relative to the page or character to which it is attached. This command is not valid for graphics boxes that are attached to paragraphs.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.

**Syntax** BoxVerticalAlignment (Alignment: *Numeric Equivalent* or *Enumerated Type*, Offset: *Numeric Expression*)

---

## Parameters

*Alignment* Numeric Equivalent or Enumerated Type: Specifies the vertical position of a box.

- 1 Top!
- 2 Center!
- 4 Bottom!
- 6 Baseline!
- 7 FullAlign!

*Offset* Numeric Expression (optional): Specifies the distance a box is offset from the page.

---

**Route** Graphics, select a box type, Position, specify vertical placement options.

---

**Example** To vertically center a figure box that is attached to a page, the command sequence is:

```
BoxCreate (BoxType: FigureBox!)
BoxVerticalAlignment (Alignment: Center!)
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ ?BoxVerticalAlignment ♦ ?BoxVerticalOffset

---

# BoxVerticalPosition

**Purpose** BoxVerticalPosition specifies the absolute vertical position of a graphics box from the top of a page or paragraph. This command is not valid for boxes attached to characters.

IMPORTANT: To align a box relative to a page or character to which it is attached, use BoxVerticalAlignment.

IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.



|                   |                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | BoxVerticalPosition (Alignment: <i>Measurement Expression</i> )                                                                                                                                                         |
| <b>Parameters</b> |                                                                                                                                                                                                                         |
| <i>Alignment</i>  | Measurement Expression: Specifies the vertical position of a box.                                                                                                                                                       |
| <b>Route</b>      | Graphics, select a box type, Position, select Put Box on Current Page and select Top of Page from the Vertical pop-up list or select Put Box in Current Paragraph, enter a measurement in the Vertical Place entry box. |
| <b>Example</b>    | To position a text box three inches from the top of the page, the command sequence is:<br><br><pre> TextBoxCreate () BoxAttachTo (Anchor: Page!) BoxVerticalPosition (Alignment: 3.0") BoxEnd (State: Save!) </pre>     |
| <b>See Also</b>   | ♦ BoxCreate ♦ BoxEdit ♦ BoxEnd ♦ BoxVerticalAlignment ♦ TextBoxCreate                                                                                                                                                   |

---

## BoxWidth

|                   |                                                                                                                                                                                                                                                                                |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | BoxWidth sets the box width or allows WordPerfect to automatically set the width based on the box contents.<br><br>IMPORTANT: This command must be preceded by a command which specifies a box, such as BoxCreate, and followed by a closing command such as BoxEnd.           |
| <b>Syntax</b>     | BoxWidth (Width: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> or <i>Numeric Expression</i> )                                                                                                                                                                            |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                |
| <i>Width</i>      | Numeric Equivalent or Enumerated Type: Specifies the width of a graphics box. Select a numeric equivalent or enumerated type to automatically set the width based on the width of the box contents. Use a numeric expression if you want to set the width.<br><br>2 AutoWidth! |
| <b>Route</b>      | Graphics, select a box type, Size, select a width option.                                                                                                                                                                                                                      |
| <b>Example</b>    | To set a figure box width to three inches, the command sequence is:<br><br><pre> BoxCreate (BoxType: FigureBox!) BoxWidth (Width: 3") BoxEnd (State: Save!) </pre>                                                                                                             |

---

**See Also**

♦ `BoxCreate` ♦ `BoxEdit` ♦ `BoxEnd` ♦ `?BoxWidth`

---

## BoxWrapDlg

---

**Purpose** `BoxWrapDlg` displays the Wrap Text dialog box.

---

**Syntax** `BoxWrapDlg ()`

---

**Route** Graphics, select a box or box type, Wrap button.

---

## BulletDlg

---

**Purpose** `BulletDlg` displays the Bullets & Numbers dialog box.

---

**Syntax** `BulletDlg ()`

---

**Route** Insert, Bullets & Numbers.

---

## ButtonBarCreateDlg

---

**Purpose** `ButtonBarCreateDlg` displays the Create Button Bar dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** `ButtonBarCreateDlg ()`

---

## ButtonBarEditDlg

---

**Purpose** `ButtonBarEditDlg` displays the Button Bar Editor-WordPerfect dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**                      ButtonBarEditDlg ()

---

## ButtonBarOptionsDlg

---

**Purpose**                      ButtonBarOptionsDlg displays the Button Bar Options dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**                      ButtonBarOptionsDlg ()

---

## ButtonBarSelect

---

**Purpose**                      ButtonBarSelect selects a specified Button Bar.

---

**Syntax**                      ButtonBarSelect (ButtonBarName: *Character Expression*)

---

### Parameters

*ButtonBarName*              Character Expression: Specifies a Button Bar to select.

---

**Route**                      File, Preferences, Button Bar icon, highlight a Button Bar, Select.

---

**Example**                    To select a Button Bar named MyMacros, the command is:

ButtonBarSelect (ButtonBarName: "MyMacros")

---

## ButtonBarSetFont

---

**Purpose**                      ButtonBarSetFont specifies a Button Bar font.

---

**Syntax**                      ButtonBarSetFont (Font: *Character Expression*; Size: *Numeric Expression*)

---

### Parameters

*Font*                          Character Expression: Specifies the name of a Button Bar font.

---

|                 |                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| <i>Size</i>     | Numeric Expression: Specifies the point size of a Button Bar font.                                                                 |
| <b>Route</b>    | File, Preferences, Button Bar icon, select a Button Bar, Options, select a font face, select a font size.                          |
| <b>Example</b>  | To specify 14pt Courier as the current Button Bar font, the command is:<br><pre>ButtonBarSetFont (Font: "Courier"; Size: 14)</pre> |
| <b>See Also</b> | ♦ ButtonBarSetLocation ♦ ButtonBarSetRows ♦ ButtonBarSetStyle                                                                      |

---

## ButtonBarSetLocation

|                   |                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ButtonBarSetLocation specifies a Button Bar location.                                                                                                              |
| <b>Syntax</b>     | ButtonBarSetLocation: (Location: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                             |
| <b>Parameters</b> |                                                                                                                                                                    |
| <i>Location</i>   | Numeric Equivalent or Enumerated Type: Specifies the location of the current Button Bar.<br><pre>256  Palette! 512  Left! 1024 Top! 2048 Right! 4096 Bottom!</pre> |
| <b>Route</b>      | File, Preferences, Button Bar icon, select a Button Bar, Options, select a location.                                                                               |
| <b>Example</b>    | To display a Button Bar on the left side of the screen, the command is:<br><pre>ButtonBarSetLocation (Location: Left!)</pre>                                       |
| <b>See Also</b>   | ♦ ButtonBarSetFont ♦ ButtonBarSetRows ♦ ButtonBarSetStyle                                                                                                          |

---

## ButtonBarSetRows

|                |                                                                        |
|----------------|------------------------------------------------------------------------|
| <b>Purpose</b> | ButtonBarSetRows specifies the number of rows to show on a Button Bar. |
|----------------|------------------------------------------------------------------------|

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | ButtonBarSetRows (Rows: <i>Numeric Expression</i> )                                                                   |
| <b>Parameters</b> |                                                                                                                       |
| <i>Rows</i>       | Numeric Expression: Specifies the number of rows to show on the current Button Bar.                                   |
| <b>Route</b>      | File, Preferences, Button Bar icon, select a Button Bar, Options, specify a maximum number of rows/columns to show.   |
| <b>Example</b>    | To specify a maximum of two rows to show on the current Button Bar, the command is:<br><br>ButtonBarSetRows (Rows: 2) |
| <b>See Also</b>   | ♦ ButtonBarSetFont ♦ ButtonBarSetLocation ♦ ButtonBarSetStyle                                                         |

---

## ButtonBarSetStyle

|                   |                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ButtonBarSetStyle specifies a Button Bar style.                                                                                                                                                |
| <b>Syntax</b>     | ButtonBarSetStyle (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                |
| <b>Parameters</b> |                                                                                                                                                                                                |
| <i>Style</i>      | Numeric Equivalent or Enumerated Type: Specifies the current Button Bar style.<br><br><ol style="list-style-type: none"> <li>1 Picture!</li> <li>2 Text!</li> <li>3 PictureAndText!</li> </ol> |
| <b>Route</b>      | File, Preferences, Button Bar icon, select a Button Bar, Options, select Text, Picture, or Picture and Text.                                                                                   |
| <b>Example</b>    | To specify a picture-style Button Bar, the command is:<br><br>ButtonBarSetStyle (Style: Picture!)                                                                                              |
| <b>See Also</b>   | ♦ ButtonBarSetFont ♦ ButtonBarSetLocation ♦ ButtonBarSetRows                                                                                                                                   |

---

## ButtonBarShow

|                   |                                                                            |
|-------------------|----------------------------------------------------------------------------|
| <b>Purpose</b>    | ButtonBarShow specifies whether to display the Button Bar.                 |
| <b>Syntax</b>     | ButtonBarShow(State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                            |
| <i>State</i>      | Numeric Equivalent or Enumerated Type:<br>0 Off!<br>1 On!                  |
| <b>Route</b>      | View, Button Bar.                                                          |
| <b>Example</b>    | To display a Button Bar, the command is:<br><br>ButtonBarShow (State: On!) |

---

## ButtonBarShowToggle

|                 |                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ButtonBarShowToggle specifies whether to show a Button Bar.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use ButtonBarShow |
| <b>Syntax</b>   | ButtonBarShowToggle ()                                                                                                                                                                                          |
| <b>See Also</b> | ♦ ButtonBarShow                                                                                                                                                                                                 |

---

## CaseToggle

|                |                                                                                                                                                                             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | CaseToggle changes selected text to uppercase or lowercase letters.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | CaseToggle ()                                                                                                                                                               |

---

# Center

|                 |                                   |
|-----------------|-----------------------------------|
| <b>Purpose</b>  | Center centers one line of text.  |
| <b>Syntax</b>   | Center ()                         |
| <b>Route</b>    | Layout, Line, Center.             |
| <b>See Also</b> | ♦ CenterCurrentPage ♦ CenterPages |

---

# CenterCurrentPage

|                   |                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | CenterCurrentPage specifies whether to vertically center the current page.                                                         |
| <b>Syntax</b>     | CenterCurrentPage (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                    |
| <b>Parameters</b> |                                                                                                                                    |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to vertically center the contents of the current page.<br>0 Off!<br>1 On! |
| <b>Route</b>      | Layout, Page, Center, Current Page.                                                                                                |
| <b>Example</b>    | To vertically center the current page, the command is:<br>CenterCurrentPage (State: On!)                                           |
| <b>See Also</b>   | ♦ CenterPages ♦ ?CenterCurrentPage                                                                                                 |

---

# CenterPageDlg

|                |                                                       |
|----------------|-------------------------------------------------------|
| <b>Purpose</b> | CenterPageDlg displays the Center Page(s) dialog box. |
| <b>Syntax</b>  | CenterPageDlg ()                                      |
| <b>Route</b>   | Layout, Page, Center.                                 |

---

# CenterPages

---

**Purpose** CenterPages specifies whether the current page and all subsequent pages center vertically.

---

**Syntax** CenterPages (State: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Designates whether to center all pages vertically in a current document.

0 Off!  
1 On!

---

**Route** Layout, Page, Center, Current and Subsequent Pages.

---

**Example** To vertically center all pages in the current document, the command is:

CenterPages (State: On!)

---

**See Also** ♦ Center ♦ CenterCurrentPage ♦ ?CenterPages

---

# ChapterNumber

---

**Purpose** ChapterNumber specifies a chapter number.

---

**Syntax** ChapterNumber (Chapter: *Numeric Expression*)

---

## Parameters

*Chapter* Numeric Expression: Specifies a chapter number.

---

**Route** Layout, Page, Numbering, Value, New Chapter Number, specify number.

---

**Example** To specify chapter 5, the command is:

ChapterNumber (Chapter: 5)

---

**See Also** ♦ ChapterNumberDecrement ♦ ChapterNumberDisplay  
♦ ChapterNumberIncrement ♦ ChapterNumberMethod ♦ ?ChapterNumber



---

## ChapterNumberDecrement

|                 |                                                                                          |
|-----------------|------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ChapterNumberDecrement decreases the chapter number by one.                              |
| <b>Syntax</b>   | ChapterNumberDecrement ()                                                                |
| <b>Route</b>    | Layout, Page, Numbering, Value, specify a negative existing chapter number.              |
| <b>See Also</b> | ♦ ChapterNumber ♦ ChapterNumberDisplay ♦ ChapterNumberIncrement<br>♦ ChapterNumberMethod |

---

## ChapterNumberDisplay

|                 |                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ChapterNumberDisplay displays the chapter number at the insertion point.                                          |
| <b>Syntax</b>   | ChapterNumberDisplay ()                                                                                           |
| <b>Route</b>    | Layout, Page, Numbering, Value, select Insert and Display at Insertion Point from the Chapter Settings group box. |
| <b>See Also</b> | ♦ ChapterNumber ♦ ChapterNumberDecrement ♦ ChapterNumberIncrement<br>♦ ChapterNumberMethod                        |

---

## ChapterNumberIncrement

|                 |                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ChapterNumberIncrement increases the chapter number by one.                                                                                      |
| <b>Syntax</b>   | ChapterNumberIncrement ()                                                                                                                        |
| <b>Route</b>    | Layout, Page, Numbering, Value, select Increase/Decrease Existing Chapter Number from the Chapter Settings group box, specify a positive number. |
| <b>See Also</b> | ♦ ChapterNumber ♦ ChapterNumberDecrement ♦ ChapterNumberDisplay<br>♦ ChapterNumberMethod                                                         |

---

## ChapterNumberMethod

---

|                   |                                                                                                                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ChapterNumberMethod specifies the method of numbering a chapter.                                                                                                                                                                                        |
| <b>Syntax</b>     | ChapterNumberMethod (Method: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                      |
| <b>Parameters</b> |                                                                                                                                                                                                                                                         |
| <i>Method</i>     | Numeric Equivalent or Enumerated Type: Designates the numbering method for a current chapter.<br><ul style="list-style-type: none"><li>0 Numbers!</li><li>1 LowerLetters!</li><li>2 UpperLetters!</li><li>3 LowerRoman!</li><li>4 UpperRoman!</li></ul> |
| <b>Route</b>      | Layout, Page, Numbering, Options, Chapter, select method.                                                                                                                                                                                               |
| <b>Example</b>    | To select uppercase roman numerals as the numbering method, the command is:<br><br>ChapterNumberMethod (Method: UpperRoman!)                                                                                                                            |
| <b>See Also</b>   | ♦ ChapterNumber ♦ ChapterNumberDecrement ♦ ChapterNumberDisplay<br>♦ ChapterNumberIncrement ♦ ?ChapterNumberMethod                                                                                                                                      |

---

## Char

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Char prompts the user for a keystroke. The ASCII numeric equivalent of the keystroke is then assigned to the variable specified in the Character parameter.<br><br>IMPORTANT: Only one character is accepted and assigned to the variable. If you enter more than one character, only the first is accepted.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | Char (Character: <i>Variable</i> , Prompt: <i>Character Expression</i> )                                                                                                                                                                                                                                                                                                                                             |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>Character</i>  | Variable: Specifies a variable to which the ASCII numeric equivalent of a keystroke is assigned. Any variable is valid for this parameter.                                                                                                                                                                                                                                                                           |

|                 |                                                                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Prompt</i>   | Character Expression: Specifies a prompt to display above the entry field.                                                                                         |
| <b>Example</b>  | To get the ASCII numeric equivalent of the letter A and assign it to the variable vNum, the command is:<br><br>Char (Character: vNum; Prompt: "Enter a character") |
| <b>See Also</b> | ♦ SWITCH ♦ GETNUMBER ♦ GETSTRING ♦ IF                                                                                                                              |

---

## CharMappingDlg

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | CharMappingDlg displays the Document Character Map dialog box. |
| <b>Syntax</b>  | CharMappingDlg ()                                              |
| <b>Route</b>   | Layout, Document, Character Mapping.                           |

---

## ClearDoc

|                |                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ClearDoc closes the current document without saving and leaves the current window open.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | ClearDoc ()                                                                                                                                                                                     |

---

## ClipboardAppendTo

|                 |                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ClipboardAppendTo adds the contents of the current document to the clipboard.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | ClipboardAppendTo ()                                                                                                                                                                  |
| <b>See Also</b> | ♦ Append ♦ ClipboardRetrieve ♦ ClipboardSaveTo                                                                                                                                        |

---

## ClipboardRetrieve

|                 |                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ClipboardRetrieve inserts clipboard contents at the insertion point.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | ClipboardRetrieve ()                                                                                                                                                     |
| <b>See Also</b> | ♦ Append ♦ ClipboardAppendTo ♦ ClipboardSaveTo                                                                                                                           |

---

## ClipboardSaveTo

|                 |                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ClipboardSaveTo saves selected text or graphics to the clipboard, replacing existing clipboard contents.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.<br>IMPORTANT: If you do not select text or graphics before using this command, macro execution terminates or continues to a LABEL specified by ONERROR. |
| <b>Syntax</b>   | ClipboardSaveTo ()                                                                                                                                                                                                                                                                                                                                                   |
| <b>See Also</b> | ♦ Append ♦ ClipboardAppendTo ♦ ClipboardRetrieve                                                                                                                                                                                                                                                                                                                     |

---

## Close

|                   |                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Close closes the current window or text editor.                                                                                                         |
| <b>Syntax</b>     | Close(Save: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                       |
| <b>Parameters</b> |                                                                                                                                                         |
| <i>Save</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to save changes. You are prompted to save document if parameter is omitted.<br>0 No!<br>1 Yes! |

---

|                 |                                                                                          |
|-----------------|------------------------------------------------------------------------------------------|
| <b>Route</b>    | File, Close.                                                                             |
| <b>Example</b>  | To save and close the current window, the command is:<br><code>Close (Save: Yes!)</code> |
| <b>See Also</b> | ♦ <code>CloseNoSave</code>                                                               |

---

---

## CloseGraphicsControlBar

---

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| <b>Purpose</b> | <code>CloseGraphicsControlBar</code> closes the Graphics Control Bar. |
| <b>Syntax</b>  | <code>CloseGraphicsControlBar ()</code>                               |
| <b>Route</b>   | Choose Close on the Graphics Control Bar.                             |

---

---

## CloseNoSave

---

|                   |                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | <code>CloseNoSave</code> closes the active document without saving it.                                                                                                                                                            |
| <b>Syntax</b>     | <code>CloseNoSave</code> (Verify: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                           |
| <b>Parameters</b> |                                                                                                                                                                                                                                   |
| <i>Verify</i>     | Numeric Equivalent or Enumerated Type: Verifies that the user wants to close the current document without saving. You are not prompted to save the document if parameter is omitted.<br><code>0</code> No!<br><code>1</code> Yes! |
| <b>Route</b>      | Press Ctrl+Shift+F4.                                                                                                                                                                                                              |
| <b>Example</b>    | To be prompted before closing the current document, the command is:<br><code>CloseNoSave (Verify: Yes!)</code>                                                                                                                    |

---

---

# CodePage

---

**Purpose** CodePage specifies code page preferences.

---

**Syntax** CodePage (Dos: *Numeric Expression*; Windows: *Numeric Expression*; InputFile: *Numeric Expression*; OutputFile: *Numeric Expression*)

---

## Parameters

*Dos* Numeric Expression: Specifies a DOS code page setting. Use the following numbers to specify code page settings:

- 333 Europa
- 437 US IBM Default
- 710 Arabic
- 850 PC Multilingual
- 8510 Alternate Greek
- 852 Eastern Europe
- 853 Turkish
- 860 Portugal
- 8600 Alternate Portugal
- 8601 Alternate Portugal
- 861 Icelandic
- 862 Hebrew
- 863 Canada/French
- 865 Norway
- 866 Cyrillic
- 8660 Alternate Cyrillic
- 895 Czech-Kamenicky
- 897 Hungarian

*Windows* Numeric Expression: Specifies a windows code page setting. Use the following numbers to specify code page settings:

- 1004 Windows US
- 1250 Windows Latin-2
- 1251 Windows Cyrillic
- 1252 Windows US Standard
- 1253 Windows Greek
- 1254 Windows Turkish
- ISO Latin #1
- 8859-1 ISO Latin #2
- 8859-2 ISO Latin/Arabic
- 8859-6 ISO Latin/Greek
- 8859-7 ISO Latin/Hebrew
- 8859-8

|                   |                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------|
| <i>InputFile</i>  | Numeric Expression: Specifies an input file code page setting. Use any code number listed for the DOS or Windows parameter.  |
| <i>OutputFile</i> | Numeric Expression: Specifies an output file code page setting. Use any code number listed for the DOS or Windows parameter. |

---

**Route** File, Preferences, Environment, Code Page, select options from the Code Page Settings pop-up lists.

---

**Example** To specify a Cyrillic code page setting for DOS, windows, input files, and output files, the command is:

CodePage: (Dos: 866; Windows: 1251; InputFile: 866; OutputFile: 866)

---

## CodePageDlg

---

**Purpose** CodePageDlg displays the Code Page Preferences dialog box.  
**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** CodePageDlg ()

---

## ColorAdd

---

**Purpose** ColorAdd adds a color to a color palette.  
**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** ColorAdd (PaletteName: *Character Expression*; PositionIndex: *Numeric Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; ColorName: *Character Expression*)

---

### Parameters

*PaletteName* Character Expression: Names a color palette.  
*PositionIndex* Numeric Expression: Specifies the color position in the palettes.

|                  |                                                                                           |
|------------------|-------------------------------------------------------------------------------------------|
| <i>Red</i>       | Numeric Expression: Specifies the amount of red in a color. Values range from 0 to 255.   |
| <i>Green</i>     | Numeric Expression: Specifies the amount of green in a color. Values range from 0 to 255. |
| <i>Blue</i>      | Numeric Expression: Specifies the amount of blue in a color. Values range from 0 to 255.  |
| <i>ColorName</i> | Character Expression: Names the new color.                                                |

---

**Example** To add a color named Autumn to a palette named Seasons, the command is:

```
ColorAdd (PaletteName: "Seasons"; PositionIndex: 0; Red: 251; Green: 160; Blue: 90; ColorName: "Autumn")
```

---

**See Also** ♦ ColorDelete ♦ ColorModify

---

## ColorDelete

---

**Purpose** ColorDelete removes a color from a color palette.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** ColorDelete (PaletteName: *Character Expression*; ColorName: *Character Expression*)

---

### Parameters

*PaletteName* Character Expression: Specifies a palette to delete a color from.

*ColorName* Character Expression: Specifies a color to delete.

---

**Example** To delete the color Cyan from a palette named Crayons, the command is:

```
ColorDelete (PaletteName: "Crayon"; ColorName: "Cyan")
```

---

**See Also** ♦ ColorAdd ♦ ColorModify



---

# ColorModify

---

**Purpose** ColorModify modifies and renames a color.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** ColorModify (PaletteName: *Character Expression*; OldColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; NewColorName: *Character Expression*)

---

## Parameters

*PaletteName* Character Expression: Names a color palette.  
*OldColorName* Character Expression: Specifies a color to modify.  
*Red* Numeric Expression: Specifies the amount of red in a color. Values range from 0 to 255.  
*Green* Numeric Expression: Specifies the amount of green in a color. Values range from 0 to 255.  
*Blue* Numeric Expression: Specifies the amount of blue in a color. Values range from 0 to 255.  
*NewColorName* Character Expression: Renames a modified color.

---

**Example** To modify Black in a palette named Crayons and rename the color Slate, the command is:

```
ColorModify (PaletteName: "Crayons"; OldColorName: "Black"; Red: 51; Green: 91; Blue: 91;
NewColorName: "Slate")
```

---

**See Also** ♦ ColorAdd ♦ ColorDelete

---

# ColorPaletteCreate

---

**Purpose** ColorPaletteCreate names a palette to create, usually in conjunction with other commands such as ColorAdd.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** ColorPaletteCreate (PaletteName: *Character Expression*)

---

**Parameters**

*PaletteName* Character Expression: Names a new palette.

---

**Example**

To create a palette named Hues of Blue, the command is:

ColorPaletteCreate (PaletteName: "Hues of Blue")

---

**See Also**

♦ ColorPaletteDelete ♦ ColorPaletteRename ♦ ColorPaletteSelect ♦ ColorAdd  
♦ ColorDelete ♦ ColorModify

---

---

## ColorPaletteDelete

**Purpose**

ColorPaletteDelete names a palette to delete.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

ColorPaletteDelete (PaletteName: *Character Expression*)

---

**Parameters**

*PaletteName* Character Expression: Names a palette to delete.

---

**Example**

To delete a palette named Hues of Blue, the command is:

ColorPaletteDelete (PaletteName: "Hues of Blue")

---

**See Also**

♦ ColorPaletteCreate ♦ ColorPaletteRename ♦ ColorPaletteSelect

---

---

## ColorPaletteDlg

**Purpose**

ColorPaletteDlg displays the Define Color Printing Palette dialog box.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

ColorPaletteDlg ()

---

**Route**

Graphics, Graphics Styles, Fill, Create or Edit, Palette.

---

---

## ColorPaletteRename

---

**Purpose** ColorPaletteRename renames an existing palette.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** ColorPaletteRename (OldPaletteName: *Character Expression*; NewPaletteName: *Character Expression*)

### Parameters

*OldPaletteName* Character Expression: Specifies the current name of a palette.

*NewPaletteName* Character Expression: Specifies a new name for a palette.

**Example** To change the name Jeff's Palette to Edward's Palette, the command is:  
ColorPaletteRename (OldPaletteName: "Jeff's Palette"; NewPaletteName: "Edward's Palette")

**See Also** ♦ ColorPaletteCreate ♦ ColorPaletteDelete ♦ ColorPaletteSelect

---

---

## ColorPaletteSelect

---

**Purpose** ColorPaletteSelect names a palette to select.

**Syntax** ColorPaletteSelect (PaletteName: *Character Expression*)

### Parameters

*PaletteName* Character Expression: Specifies a palette to select.

**Route** Graphics, Graphics Styles, Fill, Create or Edit, Palette, Open, select a palette.

**Example** To select a palette named Multi-Colors, the command is:  
ColorPaletteSelect (PaletteName: "Multi-Colors")

**See Also** ♦ ColorPaletteCreate ♦ ColorPaletteDelete ♦ ColorPaletteRename

---

---

## ColumnBorderDlg

|                |                                                        |
|----------------|--------------------------------------------------------|
| <b>Purpose</b> | ColumnBorderDlg displays the Column Border dialog box. |
| <b>Syntax</b>  | ColumnBorderDlg ()                                     |
| <b>Route</b>   | Layout, Columns, Border/Fill.                          |

---

## ColumnDefineDlg

|                 |                                                  |
|-----------------|--------------------------------------------------|
| <b>Purpose</b>  | ColumnDefineDlg displays the Columns dialog box. |
| <b>Syntax</b>   | ColumnDefineDlg ()                               |
| <b>Route</b>    | Layout, Columns, Define.                         |
| <b>See Also</b> | ♦ ColumnsDefiniton                               |

---

## ColumnsDefinition

|                |                                                                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ColumnsDefinition defines columns.                                                                                                                                                                                                              |
| <b>Syntax</b>  | ColumnsDefinition (ColumnsType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , VerticalSpacing: <i>Numeric Expression</i> ; {Spacing: <i>Measurement Expression</i> ; SpacingDef: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> }) |

---

### Parameters

|                        |                                                                                                                                                                                                                                                                                                                                                |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ColumnsType</i>     | Numeric Equivalent or Enumerated Type: Specifies a column type to define.<br><ul style="list-style-type: none"><li>0 Newspaper!</li><li>1 NewspaperBalanced!</li><li>2 Parallel!</li><li>3 ParallelBlockProtect!</li></ul>                                                                                                                     |
| <i>VerticalSpacing</i> | Numeric Expression: Specifies the spacing between sections of text in parallel columns.                                                                                                                                                                                                                                                        |
| <i>Spacing</i>         | Measurement Expression: Alternates between specifying the width of each column and the desired space between the columns. This parameter is always paired with the SpacingDef parameter. Enclose Spacing and SpacingDef parameters in braces ({}), separated by semicolons. For multiple columns, alternate Spacing and SpacingDef parameters. |

## *SpacingDef*

Numeric Equivalent or Enumerated Type: Specifies whether or not the width of the column or the space between columns is fixed. This parameter is always paired with the Spacing parameter. Enclose Spacing and SpacingDef parameters in braces ({}), separated by semicolons. For multiple columns, alternate Spacing and SpacingDef parameters.

- 0 NotFixed!
- 1 Fixed!

---

**Route** Layout, Columns, Define, select column type, select number of columns, specify column width for each column, specify spacing between columns, specify line spacing between rows (if applicable).

---

**Example** To define columns, the command is:

```
ColumnsDefinition (ColumnsType: Newspaper!; VerticalSpacing: 1.0; {Spacing: 3.125"; SpacingDef: NotFixed!; Spacing: 0.25"; SpacingDef: Fixed!; Spacing: 3.125"; SpacingDef: NotFixed!})
```

---

**See Also** ♦ ColumnsOff

---

## **ColumnsOff**

---

**Purpose** ColumnsOff turns Columns off.

---

**Syntax** ColumnsOff ()

---

**Route** Layout, Columns, Off.

---

**See Also** ♦ ColumnsDefinition

---

## **CommentConvert**

---

**Purpose** CommentConvert converts a comment to text.

---

**Syntax** CommentConvert ()

---

**Route** Insert, Comment, Convert to Text.

---

**See Also** ♦ CommentEdit

---

## CommentCreate

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| <b>Purpose</b>  | CommentCreate opens a new comment window.           |
| <b>Syntax</b>   | CommentCreate ()                                    |
| <b>Route</b>    | Insert, Comment, Create.                            |
| <b>See Also</b> | ♦ CommentEdit ♦ CommentConvert ♦ CommentInformation |

---

## CommentEdit

|                 |                                               |
|-----------------|-----------------------------------------------|
| <b>Purpose</b>  | CommentEdit opens the comment screen to edit. |
| <b>Syntax</b>   | CommentEdit ()                                |
| <b>Route</b>    | Insert, Comment, Edit.                        |
| <b>See Also</b> | ♦ CommentConvert                              |

---

## CommentInformation

|                |                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | CommentInformation displays the Comment Information dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | CommentInformation ()                                                                                                                                               |

---

## CommentInsertTime

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Purpose</b> | CommentInsertTime inserts the current time into a comment. |
| <b>Syntax</b>  | CommentInsertTime ()                                       |

---

|                 |                                                                                      |
|-----------------|--------------------------------------------------------------------------------------|
| <b>Route</b>    | Insert, Comment, Create or Edit, Time.                                               |
| <b>See Also</b> | ♦ CommentCreate ♦ CommentEdit ♦ CommentInsertUserInitials<br>♦ CommentInsertUserName |

---

## CommentInsertUserInitials

---

|                 |                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | CommentInsertUserInitials inserts the user initials specified in the Environment Preferences dialog box into a comment. |
| <b>Syntax</b>   | CommentInsertUserInitials ()                                                                                            |
| <b>Route</b>    | Insert, Comment, Create or Edit, Initials.                                                                              |
| <b>See Also</b> | ♦ CommentCreate ♦ CommentEdit ♦ CommentInsertTime<br>♦ CommentInsertUserName                                            |

---

## CommentInsertUserName

---

|                 |                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | CommentInsertUserName inserts the user name specified in the Environment Preferences dialog box into a comment. |
| <b>Syntax</b>   | CommentInsertUserName ()                                                                                        |
| <b>Route</b>    | Insert, Comment, Create or Edit, Name.                                                                          |
| <b>See Also</b> | ♦ CommentCreate ♦ CommentEdit ♦ CommentInsertTime<br>♦ CommentInsertUserInitials                                |

---

## ConditionalEndOfPage

---

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ConditionalEndOfPage keeps a specified number of lines together at a page break. If a page break falls between the lines, the lines move to the next page. |
| <b>Syntax</b>  | ConditionalEndOfPage (LinesTogether: <i>Numeric Expression</i> )                                                                                           |

---

---

**Parameters**

*LinesTogether* Numeric Expression: Specifies the number of lines.

---

**Route** Layout, Page, Keep Text Together, specify a number of lines.

---

**Example** To keep ten lines of text together, the command is:

`ConditionalEndOfPage (LinesTogether: 10)`

---

---

## ConvertCaseInitialCaps

---

**Purpose** ConvertCaseInitialCaps converts selected text to lowercase but capitalizes the first letter of each word.

---

**Syntax** ConvertCaseInitialCaps ()

---

**Route** Edit, Convert Case, Initial Capitals.

---

**See Also** ♦ ConvertCaseLowercase ♦ ConvertCaseUppercase

---

---

## ConvertCaseLowercase

---

**Purpose** ConvertCaseLowercase converts selected text to lowercase, except the first word of each sentence and "I" and its forms, such as "I'm" and "I've."

---

**Syntax** ConvertCaseLowercase ()

---

**Route** Edit, Convert Case, Lowercase.

---

**See Also** ♦ ConvertCaseInitialCaps ♦ ConvertCaseUppercase

---

---

## ConvertCaseUppercase

---

**Purpose** ConvertCaseUppercase converts selected text to uppercase.

---

**Syntax** ConvertCase ()

---



|                 |                                                 |
|-----------------|-------------------------------------------------|
| <b>Route</b>    | Edit, Convert Case, Uppercase.                  |
| <b>See Also</b> | ♦ ConvertCaseInitialCaps ♦ ConvertCaseLowercase |

## Copy

|                 |                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | Copy duplicates blocked text and is used with Paste to retrieve the duplicated text.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. To record a command with similar functionality, use EditCopy. |
| <b>Syntax</b>   | Copy ()                                                                                                                                                                                                                                                |
| <b>See Also</b> | ♦ EditCopy                                                                                                                                                                                                                                             |

## CounterCreate

|                |                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | CounterCreate specifies a graphics box reference number.                                                                                                                         |
| <b>Syntax</b>  | CounterCreate (CounterName: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> or <i>Character Expression</i> ; {Method: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> }) |

### Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>CounterName</i> | Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value. Use a character expression to create and name a user-defined counter.<br><ul style="list-style-type: none"> <li>0 None!</li> <li>1 FigureCounter!</li> <li>2 TableBoxCounter!</li> <li>3 TextBoxCounter!</li> <li>4 UserBoxCounter!</li> <li>5 EquationCounter!</li> </ul>             |
| <i>Method</i>      | Numeric Equivalent : Specifies a numbering method. To specify different numbering methods for each level, repeat the method parameter. Enclose the repeated parameters in braces, and place a semicolon between each method.<br><ul style="list-style-type: none"> <li>0 Numbers!</li> <li>1 LowLetter!</li> <li>2 UpLetter!</li> <li>3 LowRoman!</li> <li>4 UpRoman!</li> </ul> |

---

|                 |                                                                                                                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Insert, Other, Counter, Create, type a name.                                                                                                                                                                                                                 |
| <b>Example</b>  | To specify a graphics box reference number, and to specify lowercase letters for level 1 and lowercase roman numerals for level 2, the command is:<br><br><code>CounterCreate (CounterName: UserBoxCounter!; {Method: LowLetter!; Method: LowRoman!})</code> |
| <b>See Also</b> | ♦ CounterDelete ♦ CounterSetMethod ♦ CounterSetValue                                                                                                                                                                                                         |

---

## CounterDecrement

---

|                    |                                                                                                                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | CounterDecrement decreases a counter by one.                                                                                                                                                                       |
| <b>Syntax</b>      | CounterDecrement (CounterName: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> or <i>Character Expression</i> ; Level: <i>Numeric Expression</i> )                                                             |
| <b>Parameters</b>  |                                                                                                                                                                                                                    |
| <i>CounterName</i> | Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value.<br><br>0 None!<br>1 FigureCounter!<br>2 TableBoxCounter!<br>3 TextBoxCounter!<br>4 UserBoxCounter!<br>5 EquationCounter! |
| <i>Level</i>       | Numeric Expression: Specifies a level to decrease.                                                                                                                                                                 |
| <b>Route</b>       | Insert, Other, Counter, select a counter type, Decrease.                                                                                                                                                           |
| <b>Example</b>     | To decrease a counter by one, the command is:<br><br><code>CounterDecrement (CounterName: FigureCounter!; Level: 1)</code>                                                                                         |
| <b>See Also</b>    | ♦ CounterIncrement ♦ CounterSetValue                                                                                                                                                                               |

---

## CounterDelete

---

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | CounterDelete specifies a counter value to delete. |
|----------------|----------------------------------------------------|

---

IMPORTANT: This command accepts either a character expression, if the counter is a user-defined counter, a numeric equivalent, or an enumerated type. If a numeric equivalent or enumerated type is specified, the counter resets and does not delete.

---

**Syntax** CounterDelete (CounterName: *Numeric Equivalent or Enumerated Type or Character Expression*)

---

**Parameters**

*CounterName* Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter to delete. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TableBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

---

**Route** Insert, Other, Counter, select a counter type, Delete.

---

**Example** To delete a counter, the command is:  
CounterDelete (CounterName: "My Counter")

---

**See Also** ♦ CounterCreate

---

---

## CounterDisplay

---

**Purpose** CounterDisplay displays the counter value of a graphics box or user-defined counter.

**Syntax** CounterDisplay (CounterName: *Numeric Equivalent or Enumerated Type or Character Expression*; Level: *Numeric Expression*)

---

**Parameters**

*CounterName* Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value. This parameter can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TableBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

|                |                                                                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Level</i>   | Numeric Expression: Specifies a level of the graphics box counter to display.                                                                       |
| <b>Route</b>   | Insert, Other, Counter, select a counter, Display in Document.                                                                                      |
| <b>Example</b> | To display the counter value of a graphics box or user-defined counter, the command is:<br><br>CounterDisplay (CounterName: "My Counter"; Level: 1) |

---

## CounterDlg

|                |                                                       |
|----------------|-------------------------------------------------------|
| <b>Purpose</b> | CounterDlg displays the Counter Numbering dialog box. |
| <b>Syntax</b>  | CounterDlg ()                                         |
| <b>Route</b>   | Insert, Other, Counter.                               |

---

## CounterEditLevel

|                    |                                                                                                                                                                                                                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | CounterEditLevel specifies the number of levels a counter contains.                                                                                                                                                                                                                                                |
| <b>Syntax</b>      | CounterEditLevel (CounterName: <i>Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression</i> ; NewNumLvls: <i>Numeric Expression</i> )                                                                                                                                                   |
| <b>Parameters</b>  |                                                                                                                                                                                                                                                                                                                    |
| <i>CounterName</i> | Numeric Equivalent or Enumerated Type: Specifies a counter value type. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.<br><br>0 None!<br>1 FigureCounter!<br>2 TableBoxCounter!<br>3 TextBoxCounter!<br>4 UserBoxCounter!<br>5 EquationCounter! |
| <i>NewNumLvls</i>  | Numeric Equivalent or Enumerated Type: Specifies a counter level to display.                                                                                                                                                                                                                                       |
| <b>Route</b>       | Insert, Other, Counter, select a counter, Edit, Levels, specify a number of levels.                                                                                                                                                                                                                                |

|                 |                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b>  | To specify the number of levels a counter contains, the command is:<br><br>CounterEditLevel (CounterName: FigureCounter!; NewNumLvs: 2) |
| <b>See Also</b> | ♦ CounterEditLevelMethod ♦ CounterEditLevelName                                                                                         |

---

## CounterEditLevelMethod

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | CounterEditLevelMethod changes the numbering method for a counter level. This command is only available when the counter has more than one level.                                                                                                                                                                                                                                                                  |
| <b>Syntax</b>      | CounterEditLevelMethod (CounterName: <i>Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression</i> ; LevelNum: <i>Numeric Expression</i> ; NewMethod: <i>Numeric Equivalent or Enumerated Type or Enumerated Type</i> )                                                                                                                                                                  |
| <b>Parameters</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>CounterName</i> | Numeric Equivalent or Enumerated Type or Character Expression: Specifies a graphics box type. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type. <ul style="list-style-type: none"> <li>0 None!</li> <li>1 FigureCounter!</li> <li>2 TableBoxCounter!</li> <li>3 TextBoxCounter!</li> <li>4 UserBoxCounter!</li> <li>5 EquationCounter!</li> </ul> |
| <i>LevelNumber</i> | Numeric Expression: Specifies level number.                                                                                                                                                                                                                                                                                                                                                                        |
| <i>NewMethod</i>   | Numeric Equivalent or Enumerated Type: Specifies a display type for the counter. <ul style="list-style-type: none"> <li>0 Numbers!</li> <li>1 LowLetter!</li> <li>2 UpLetter!</li> <li>3 LowRoman!</li> <li>4 UpRoman!</li> </ul>                                                                                                                                                                                  |
| <b>Route</b>       | Insert, Other, Counter, select a counter, Edit, select a numbering method.                                                                                                                                                                                                                                                                                                                                         |
| <b>Example</b>     | To change the numbering method for a counter level, the command is:<br><br>CounterEditLevelMethod (CounterName: FigureCounter!; LevelNum: 2; NewMethod: UpLetter!)                                                                                                                                                                                                                                                 |
| <b>See Also</b>    | ♦ CounterEditMethod                                                                                                                                                                                                                                                                                                                                                                                                |

---

## CounterEditLevelName

---

**Purpose** CounterEditLevelName renames a counter level type. This command is only available if the counter has more than one level.

**Syntax** CounterEditLevelName (OldCounterName: *Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression*; OldLevelName: *Character Expression*; NewLevelName: *Character Expression*)

### Parameters

*OldCounterName* Numeric Equivalent or Enumerated Type or Character Expression: Specifies a type of counter. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TableBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

*OldLevelName* Character Expression: Specifies the old name of a graphics box counter type.

*NewLevelName* Character Expression: Specifies the new name of a graphics box counter type.

**Route** Insert, Other, Counter, select a counter with more than one level, Edit, select the text in a Level Name text box, enter a new name.

**Example** To rename a counter level type when the counter has more than one level, the command is:

```
CounterEditLevelName (OldCounterName: "My Box"; OldLevelName: "Level 1"; NewLevelName: "FirstLevel")
```

**See Also** ♦ CounterEditName

---

## CounterEditMethod

---

**Purpose** CounterEditMethod changes the numbering method for a specified counter.

**Syntax** CounterEditMethod (CounterName: *Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression*; NewMethod: *Numeric Equivalent or Enumerated Type or Enumerated Type*)

---

## Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>CounterName</i> | Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.<br><ul style="list-style-type: none"><li>0 None!</li><li>1 FigureCounter!</li><li>2 TableBoxCounter!</li><li>3 TextBoxCounter!</li><li>4 UserBoxCounter!</li><li>5 EquationCounter!</li></ul> |
| <i>NewMethod</i>   | Numeric Equivalent or Enumerated Type: Specifies a counter display type.<br><ul style="list-style-type: none"><li>0 Numbers!</li><li>1 LowLetter!</li><li>2 UpLetter!</li><li>3 LowRoman!</li><li>4 UpRoman!</li></ul>                                                                                                                                                                                     |

---

|              |                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------|
| <b>Route</b> | Insert, Other, Counter, select a counter, Edit, Single Level Method, select a numbering method. |
|--------------|-------------------------------------------------------------------------------------------------|

---

|                |                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To change the numbering method for a specified counter, the command is:<br><br>CounterEditMethod (CounterName: FigureCounter!; NewMethod: UpLetter!) |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                          |
|-----------------|--------------------------|
| <b>See Also</b> | ♦ CounterEditLevelMethod |
|-----------------|--------------------------|

---

## CounterEditName

---

|                |                                                 |
|----------------|-------------------------------------------------|
| <b>Purpose</b> | CounterEditName renames a user-created counter. |
|----------------|-------------------------------------------------|

---

|               |                                                                                                |
|---------------|------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | CounterEditName (OldName: <i>Character Expression</i> ; NewName: <i>Character Expression</i> ) |
|---------------|------------------------------------------------------------------------------------------------|

---

## Parameters

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| <i>OldName</i> | Character Expression: Specifies the name of a counter value.     |
| <i>NewName</i> | Character Expression: Specifies the new name of a counter value. |

---

|              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| <b>Route</b> | Insert, Other, Counter, select a user-created counter, Edit, Counter Name, enter a new name. |
|--------------|----------------------------------------------------------------------------------------------|

---

**Example**

To rename a user-created counter, the command is:

```
CounterEditName (OldName: "My Box"; NewName: "Shaded Box")
```

---

**See Also**

♦ CounterEditLevelName

---

---

## CounterIncrement

---

**Purpose**

CounterIncrement increases a specified counter by one.

---

**Syntax**

CounterIncrement (CounterName: *Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression*; Level: *Numeric Expression*)

---

**Parameters**

*CounterName*

Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TableBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

*Level*

Numeric Expression: Specifies a counter level to increment.

---

**Route**

Insert, Other, Counter, select a counter, Increase.

---

**Example**

To increment TableBoxCounter! by one, the command is:

```
CounterIncrement (CounterName: TableBoxCounter!; Level: 1)
```

---

**See Also**

♦ CounterDecrement

---

---

## CounterSetMethod

---

**Purpose**

CounterSetMethod specifies a counter numbering method.

---

**Syntax**

CounterSetMethod (CounterName: *Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression*; CounterLevel: *Numeric Expression*; Method: *Numeric Equivalent or Enumerated Type or Enumerated Type*)

---



---

## Parameters

### *CounterName*

Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TextBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

### *CounterLevel*

Numeric Expression: Specifies a counter level to increment.

### *Method*

Numeric Equivalent or Enumerated Type: Specifies a counter display type.

- 0 Numbers!
- 1 LowerLetter!
- 2 UpperLetter!
- 3 LowerRoman!
- 4 UpperRoman!

---

## Route

Insert, Other, Counter, select a counter, Value, Number Method, select a method.

---

## Example

To specify a counter numbering method, the command is:

```
CounterSetMethod (CounterName: TextBoxCounter!; CounterLevel: 2; Method LowerRoman!)
```

---

# CounterSetValue

---

## Purpose

CounterSetValue sets a counter value.

---

## Syntax

CounterSetValue (CounterName: *Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression*; {CounterLevel: *Numeric Expression*})

---

## Parameters

### *CounterName*

Numeric Equivalent or Enumerated Type: Specifies a counter value. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TextBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

*CounterLevel* Numeric Expression: Specifies the value of the selected counter or counter level. To change counter levels, repeat the CounterLevel parameter. Enclose the repeated parameters in braces, with a semicolon between each level.

---

**Route** Insert, Other, Counter, select a counter, Value, specify a value.

---

**Example** To set a value for FigureCounter!, the command is:

CounterSetValue (CounterName: FigureCounter!; {CounterLevel: 5})

---

## CrossRefMarkReference

---

**Purpose** CrossRefMarkReference specifies a reference mark.

---

**Syntax** CrossRefMarkReference (TargetName: *Character Expression*; RefType: *Numeric Equivalent or Enumerated Type or Enumerated Type*; CounterName: *Numeric Equivalent or Enumerated Type or Enumerated Type or Character Expression*)

---

### Parameters

*TargetName* Character Expression: Specifies a reference name.

*RefType* Numeric Equivalent or Enumerated Type: Specifies what to tie the reference to.

- 0 RefPage!
- 1 RefSecondaryPage!
- 2 RefChapter!
- 3 RefVolume!
- 4 RefParagraph!
- 5 RefFootnote!
- 6 RefEndnote!
- 7 RefBox!
- 8 RefCounter!

*CounterName* Numeric Equivalent or Enumerated Type or Character Expression: Specifies a counter value. This can be a character expression specifying a user-defined counter, a numeric equivalent, or an enumerated type.

- 0 None!
- 1 FigureCounter!
- 2 TableBoxCounter!
- 3 TextBoxCounter!
- 4 UserBoxCounter!
- 5 EquationCounter!

---

|              |                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------|
| <b>Route</b> | Tools, Cross-Reference, Reference, specify a reference type, enter a target name, Mark Reference. |
|--------------|---------------------------------------------------------------------------------------------------|

---

|                |                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify a reference mark, the command is:<br><code>CrossRefMarkReference (TargetName: "Next"; RefType: RefSecondaryPage!)</code> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|

---

---

## CrossRefMarkTarget

---

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | CrossRefMarkTarget names a cross-reference target. |
|----------------|----------------------------------------------------|

---

|               |                                                            |
|---------------|------------------------------------------------------------|
| <b>Syntax</b> | CrossRefMarkTarget (TagName: <i>Character Expression</i> ) |
|---------------|------------------------------------------------------------|

---

### Parameters

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| <i>TagName</i> | Character Expression: Specifies the name of a cross-reference target. |
|----------------|-----------------------------------------------------------------------|

---

|              |                                                                   |
|--------------|-------------------------------------------------------------------|
| <b>Route</b> | Tools, Cross-Reference, Target, enter a target name, Mark Target. |
|--------------|-------------------------------------------------------------------|

---

|                |                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To name a cross-reference target, the command is:<br><code>CrossRefMarkTarget (TagName: "Index")</code> |
|----------------|---------------------------------------------------------------------------------------------------------|

---

---

## CustomBoxDlg

---

|                |                                                  |
|----------------|--------------------------------------------------|
| <b>Purpose</b> | CustomBoxDlg displays the Custom Box dialog box. |
|----------------|--------------------------------------------------|

---

|               |                 |
|---------------|-----------------|
| <b>Syntax</b> | CustomBoxDlg () |
|---------------|-----------------|

---

|              |                       |
|--------------|-----------------------|
| <b>Route</b> | Graphics, Custom Box. |
|--------------|-----------------------|

---

---

## DateCode

---

|                |                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DateCode inserts a code to display the current date at the insertion point. The date code updates to the current system date whenever the document is retrieved or reformatted. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

|                 |                          |
|-----------------|--------------------------|
| <b>Syntax</b>   | DateCode ()              |
| <b>Route</b>    | Insert, Date, Date Code. |
| <b>See Also</b> | ♦ DateText               |

---

## DateFormat

---

|                |                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DateFormat specifies a date format through DateCode or DateText.<br>IMPORTANT: This command must be recorded. |
|----------------|---------------------------------------------------------------------------------------------------------------|

---

|               |                                                             |
|---------------|-------------------------------------------------------------|
| <b>Syntax</b> | DateFormat (DateFormatString: <i>Character Expression</i> ) |
|---------------|-------------------------------------------------------------|

---

### Parameters

|                         |                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------|
| <i>DateFormatString</i> | Character Expression: Specifies an order of day, month, year codes, and time codes. |
|-------------------------|-------------------------------------------------------------------------------------|

---

|              |                                              |
|--------------|----------------------------------------------|
| <b>Route</b> | Insert, Date, Date Format, specify a format. |
|--------------|----------------------------------------------|

---

|                |                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify a format such as June 11, 1964, the command is:<br>DateFormat (DateFormatString: "Month Name Day, Year (4 Digit)") |
|----------------|-------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                 |
|-----------------|-----------------|
| <b>See Also</b> | ♦ DateFormatDlg |
|-----------------|-----------------|

---

## DateFormatDlg

---

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| <b>Purpose</b> | DateFormatDlg displays the Document Date/Time Format dialog box. |
|----------------|------------------------------------------------------------------|

---

|               |                  |
|---------------|------------------|
| <b>Syntax</b> | DateFormatDlg () |
|---------------|------------------|

---

|              |                            |
|--------------|----------------------------|
| <b>Route</b> | Insert, Date, Date Format. |
|--------------|----------------------------|

---

|                 |              |
|-----------------|--------------|
| <b>See Also</b> | ♦ DateFormat |
|-----------------|--------------|

---

---

## DateText

---

**Purpose** DateText inserts the system date at the insertion point.

---

**Syntax** DateText ()

---

**Route** Insert, Date, Date Text.

---

**See Also** ♦ DateCode

---

---

## DDEChangeUpdateMode

---

**Purpose** DDEChangeUpdateMode specifies whether to update a DDE link manually or automatically.

---

**Syntax** DDEChangeUpdateMode (LinkName: *Character Expression*; UpdateMode: *Numeric Equivalent or Enumerated Type* or *Enumerated Type*)

---

### Parameters

*LinkName* Character Expression: Specifies a DDE link.

*UpdateMode* Numeric Equivalent or Enumerated Type: Specifies whether to update a link manually or automatically.

- 0 Manual!
- 1 Automatic!

---

## DDECreateLink

---

**Purpose** DDECreateLink establishes a link between a WordPerfect document and a file in another application. As data changes in the source file, the changes also appear in the WordPerfect document.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** DDECreateLink (LinkName: *Character Expression*; Source: *Character Expression*; UpdateMode: *Numeric Equivalent or Enumerated Type* or *Enumerated Type*; Format: *Numeric Equivalent or Enumerated Type* or *Enumerated Type*)

---

---

## Parameters

|                   |                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>LinkName</i>   | Character Expression: Specifies a link name.                                                                                                                                                                                                                                                                          |
| <i>Source</i>     | Character Expression: Specifies the application, file, and optionally, an item within the file that is linked to a WordPerfect document. The application and filename are separated by a vertical bar (!). If an item (a section of a file) is specified, a second vertical bar separates the item from the filename. |
| <i>UpdateMode</i> | Numeric Equivalent: Specifies whether a link is updated manually or automatically.<br>0 Manual!<br>1 Automatic!                                                                                                                                                                                                       |
| <i>Format</i>     | Numeric Equivalent or Enumerated Type: Specifies a format for linked information.<br>0 WP6DOC!<br>1 WPDOC!<br>2 WP6TEXT!<br>3 WPG20!<br>4 RTF!<br>5 TEXT!<br>6 OEMTEXT!<br>7 DIB!<br>8 BITMAP!<br>9 METAFILEPICT!                                                                                                     |

---

## DDEDeleteLink

---

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DDEDeleteLink removes a DDELink.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                                                        |
|---------------|--------------------------------------------------------|
| <b>Syntax</b> | DDEDeleteLink (LinkName: <i>Character Expression</i> ) |
|---------------|--------------------------------------------------------|

---

### Parameters

|                 |                                                   |
|-----------------|---------------------------------------------------|
| <i>LinkName</i> | Character Expression: Specifies a link to remove. |
|-----------------|---------------------------------------------------|

---

## DDEEditLink

---

|                |                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DDEEditLink changes the properties of a DDE Link.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Syntax** DDEEditLink (LinkName: *Character Expression*; NewName: *Character Expression*; NewSource: *Character Expression*)

---

**Parameters**

*LinkName* Character Expression: Specifies a link to edit.  
*NewName* Character Expression: Specifies a new link name.  
*NewSource* Character Expression: Specifies a new source file.

---

---

## DDEPasteLink

---

**Purpose** DDEPasteLink establishes a link between a WordPerfect document and a file in another application. As data changes in the source file, the changes also appear in the WordPerfect document. After copying a selection in another application to the Clipboard, use DDEPasteLink to link it to a WordPerfect document.

---

**Syntax** DDEPasteLink (Format: *Numeric Equivalent or Enumerated Type or Enumerated Type*)

---

**Parameters**

*Format* Numeric Equivalent or Enumerated Type: Specifies a format for pasted information.

- 0 WP6DOC!
- 1 WPDOC!
- 2 WP6TEXT!
- 3 WPG20!
- 4 RTF!
- 5 TEXT!
- 6 OEMTEXT!
- 7 DIB!
- 8 BITMAP!
- 9 METAFILEPICT!

---

---

## DDEUpdateLink

---

**Purpose** DDEUpdateLink updates a specified DDELink.

---

---

**Syntax** DDEUpdateLink (LinkName: *Character Expression*)

---

**Parameters**

*LinkName* Character Expression: Specifies a link to update.

---

---

## DecimalAlignmentCharacter

---

**Purpose** DecimalAlignmentCharacter specifies the ASCII numeric equivalent for a decimal character.

---

**Syntax** DecimalAlignmentCharacter (Character: *Numeric Expression*)

---

**Parameters**

*Character* Numeric Expression: Specifies a decimal separator character.

---

**Route** Layout, Line, Tab Set, enter a character in the Align Character text box.

---

**Example** To use a comma as a decimal alignment character, the command is:

DecimalAlignmentCharacter (Character: 44)

---

**See Also** ♦ ?DecimalAlignmentCharacter

---

---

## DelayCodes

---

**Purpose** DelayCodes postpones the effects of codes for one or more pages.

---

**Syntax** DelayCodes (PagesToDelay: *Numeric Expression*)

---

**Parameters**

*PagesToDelay* Numeric Expression: Specifies a number of pages to postpone codes.

---

**Route** Layout, Page, Delay Codes, enter number of pages.

---

**Example** To postpone codes for two pages, the command is:

DelayCodes (PagesToDelay: 2)

---



---

## DelayCodesDlg

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | DelayCodesDlg displays the Delay Codes dialog box. |
| <b>Syntax</b>  | DelayCodesDlg ()                                   |
| <b>Route</b>   | Layout, Page, Delay Codes.                         |

---

## DeleteAppend

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| <b>Purpose</b> | DeleteAppend deletes a selection and appends it to the Clipboard. |
| <b>Syntax</b>  | DeleteAppend ()                                                   |
| <b>Route</b>   | Select text, Edit, Append, press Delete.                          |

---

## DeleteCharNext

|                 |                                                                           |
|-----------------|---------------------------------------------------------------------------|
| <b>Purpose</b>  | DeleteCharNext deletes the character to the right of the insertion point. |
| <b>Syntax</b>   | DeleteCharNext ()                                                         |
| <b>Route</b>    | Press Delete.                                                             |
| <b>See Also</b> | ♦ DeleteCharPrevious                                                      |

---

## DeleteCharPrevious

|                 |                                                                              |
|-----------------|------------------------------------------------------------------------------|
| <b>Purpose</b>  | DeleteCharPrevious deletes the character to the left of the insertion point. |
| <b>Syntax</b>   | DeleteCharPrevious ()                                                        |
| <b>Route</b>    | Press Backspace.                                                             |
| <b>See Also</b> | ♦ DeleteCharNext                                                             |

---

## DeleteToBeginningOfWord

|                 |                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | DeleteToBeginningOfWord deletes text and codes from the insertion point to the beginning of the current word.<br><br>IMPORTANT: The insertion point must be positioned within a word.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | DeleteToBeginningOfWord ()                                                                                                                                                                                                                                                                    |
| <b>Route</b>    | Ctrl+Shift+Left Arrow, Delete.                                                                                                                                                                                                                                                                |
| <b>See Also</b> | ♦ DeleteToEndOfWord                                                                                                                                                                                                                                                                           |

---

## DeleteToEndOfLine

|                |                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DeleteToEndOfLine deletes text and codes from the insertion point to the end of the current line.<br><br>IMPORTANT: This command does not delete hard page breaks. |
| <b>Syntax</b>  | DeleteToEndOfLine ()                                                                                                                                               |
| <b>Route</b>   | Press Ctrl+Delete.                                                                                                                                                 |

---

## DeleteToEndOfPage

|                |                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DeleteToEndOfPage deletes text and codes from the insertion point to the end of the current page. |
| <b>Syntax</b>  | DeleteToEndOfPage ()                                                                              |
| <b>Route</b>   | Press Ctrl+Shift+Delete.                                                                          |

---

## DeleteToEndOfWord

|                 |                                                                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | DeleteToEndOfWord deletes text and codes from the insertion point to the beginning of the following word.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | DeleteToEndOfWord ()                                                                                                                                                                                              |
| <b>Route</b>    | Press Ctrl+Shift+Right Arrow, Delete.                                                                                                                                                                             |
| <b>See Also</b> | ♦ DeleteToBeginningOfWord                                                                                                                                                                                         |

---

## DeleteWord

|                |                                               |
|----------------|-----------------------------------------------|
| <b>Purpose</b> | DeleteWord deletes the current word or space. |
| <b>Syntax</b>  | DeleteWord ()                                 |
| <b>Route</b>   | Press Ctrl+Backspace.                         |

---

## DirectoryCreate

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | DirectoryCreate creates a new directory.                                                        |
| <b>Syntax</b>     | DirectoryCreate (Directory: <i>Character Expression</i> )                                       |
| <b>Parameters</b> |                                                                                                 |
| <i>Directory</i>  | Character Expression: Specifies the full path and name of a directory to create.                |
| <b>Route</b>      | File, Open, File Options, Create Directory.                                                     |
| <b>Example</b>    | To create a directory named Data, the command is:<br><br>DirectoryCreate (Directory: "C:\DATA") |

---

# Display

---

|                   |                                                                                                                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Display specifies whether macro execution is displayed while a macro is running.<br>IMPORTANT: A macro runs faster with display off. A "Please Wait" message appears on the status line.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | Display (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                        |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                              |
| <i>State</i>      | Numeric Equivalent:<br>0 Off!<br>1 On!                                                                                                                                                                                                                                                       |
| <b>Route</b>      | This command is not recordable. To use this command, you must type it into the macro.                                                                                                                                                                                                        |
| <b>Example</b>    | To turn on Display during macro execution, the command is:<br>Display (State: On!)                                                                                                                                                                                                           |
| <b>See Also</b>   | ♦ InhibitInput                                                                                                                                                                                                                                                                               |

---

# DisplayZoom

---

|                   |                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | DisplayZoom specifies the display of a current document screen.                                     |
| <b>Syntax</b>     | DisplayZoom (Percent: <i>Numeric Expression</i> )                                                   |
| <b>Parameters</b> |                                                                                                     |
| <i>Percent</i>    | Numeric Expression: Specifies a zoom percentage 25 and 400.                                         |
| <b>Route</b>      | View, Zoom, select a percentage or specify a percentage in the other entry box.                     |
| <b>Example</b>    | To specify the display of the current document at 50%, the command is:<br>DisplayZoom (Percent: 50) |

---

---

# DocCompare

---

|                   |                                                                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | DocCompare compares the current document with a specified document marking changes with Redline and Strikeout.                                                                                                                                         |
| <b>Syntax</b>     | DocCompare (Filename: <i>Character Expression</i> ; CompFlags: <i>Numeric Equivalent or Enumerated Type</i> )                                                                                                                                          |
| <b>Parameters</b> |                                                                                                                                                                                                                                                        |
| <i>Filename</i>   | Character Expression: Specifies the name of the original document.                                                                                                                                                                                     |
| <i>CompFlags</i>  | Numeric Equivalent or Enumerated Type: Specifies the scope of comparison between the original document and the current document.<br><ul style="list-style-type: none"><li>0 Phrase!</li><li>1 Word!</li><li>2 Sentence!</li><li>3 Paragraph!</li></ul> |
| <b>Route</b>      | File, Compare Document, Add Markings, enter a filename, select a comparison level.                                                                                                                                                                     |
| <b>Example</b>    | To compare the current document with a file named ORIGINAL, marking only entirely new or deleted sentences Redline and Strikeout, the command is:<br><br>DocCompare (Filename: "ORIGINAL"; CompFlags: Sentence!)                                       |
| <b>See Also</b>   | ♦ DocCompareAddMarkingsDlg ♦ DocCompareRemoveMarkingsDlg                                                                                                                                                                                               |

---

---

# DocCompareAddMarkingsDlg

---

|                 |                                                                |
|-----------------|----------------------------------------------------------------|
| <b>Purpose</b>  | DocCompareAddMarkingsDlg displays the Add Markings dialog box. |
| <b>Syntax</b>   | DocCompareAddMarkingsDlg ()                                    |
| <b>Route</b>    | File, Compare Document, Add Markings.                          |
| <b>See Also</b> | ♦ DocCompareRemoveMarkingsDlg                                  |

---

---

## DocCompareRemoveMarkingsDlg

|                 |                                                                      |
|-----------------|----------------------------------------------------------------------|
| <b>Purpose</b>  | DocCompareRemoveMarkingsDlg displays the Remove Markings dialog box. |
| <b>Syntax</b>   | DocCompareRemoveMarkingsDlg ()                                       |
| <b>Route</b>    | File, Compare Document, Remove Markings.                             |
| <b>See Also</b> | ♦ DocCompare ♦ DocCompareAddMarkingsDlg                              |

---

## DocCompareRemoveRedline

|                |                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DocCompareRemoveRedline either removes Strikeout and Redline codes as well as Strikeout text; or removes Strikeout codes and text, retaining Redline codes. |
| <b>Syntax</b>  | DocCompareRemoveRedline (Flag: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                        |

### Parameters

|             |                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Flag</i> | Numeric Equivalent or Enumerated Type: Specifies whether to remove text contained within Strikeout and Redline codes.<br>0 RemoveAll!<br>1 LeaveRedline! |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|

|              |                                                                   |
|--------------|-------------------------------------------------------------------|
| <b>Route</b> | File, Compare Document, Remove Markings, specify codes to remove. |
|--------------|-------------------------------------------------------------------|

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Example</b> | To remove all Strikeout and Redline codes, the command is: |
|----------------|------------------------------------------------------------|

DocCompareRemoveRedline (Flag: RemoveAll!)

|                 |              |
|-----------------|--------------|
| <b>See Also</b> | ♦ DocCompare |
|-----------------|--------------|

---

## DocInitialFont

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | DocInitialFont sets the default font for the current document. |
|----------------|----------------------------------------------------------------|

|               |                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | DocInitialFont (Name: <i>Character Expression</i> ; PointSize: <i>Numeric Expression</i> ; Family: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Attributes: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Weight: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Width: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Source: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Type: *Numeric Equivalent* or *Enumerated Type*; CharacterSet: *Numeric Equivalent* or *Enumerated Type*)

*Name*

Character Expression: Specifies a font name.

*PointSize*

Numeric Expression: Specifies a font size.

*Family*

Numeric Equivalent or Enumerated Type: Specifies a font family.

256 FamilyITCBenquiat!  
2304 FamilyTimesRoman!  
4352 FamilyTCGaramond!  
6400 FamilyCenturySchoolbook!  
8448 FamilyBodoni!  
10496 FamilyITCLubalinGraph!  
12544 FamilyAlbertus!  
14592 FamilyCourier!  
16640 FamilyHelvetica!  
18688 FamilyITCRonda!  
20736 FamilyMicroStyle!  
22784 FamilyOptima!  
24832 FamilyLetterGothic!  
26880 FamilyBrush!  
28928 FamilyParkAvenue!  
30976 FamilySignetRoundhand!  
33024 FamilyTekton!  
35072 FamilyUnical!  
37120 FamilyOldEnglish!  
39168 FamilyITCZapfDingbats!  
41216 FamilyBroadway!  
43264 FamilyComputer!  
45312 FamilyPTBarnum!  
47360 FamilyPonderosa!  
49408 FamilyMadrone!  
51456 FamilyHobo!  
53504 FamilyRevue!  
55552 FamilyAestheticOrnamented!

*Attributes*

Numeric Equivalent or Enumerated Type: Specifies font attributes.

0 FontMatchNormal!  
1 FontMatchItalic!  
2 FontMatchOutline!  
4 FontMatchShadow!  
8 FontMatchSmallCaps!

*Weight*

Numeric Equivalent or Enumerated Type: Specifies a font weight.

0 WeightUnknown!  
1 WeightUltraThin!  
8 WeightUltraLight!

16 WeightExtraThin!  
 24 WeightThin!  
 32 WeightExtraLight!  
 48 WeightLight!  
 64 WeightDemiLight!  
 80 WeightSemiLight!  
 88 WeightBook!  
 96 WeightRegular!  
 104 WeightRoman!  
 112 WeightMedium!  
 128 WeightDemiBold!  
 136 WeightSemiBold!  
 144 WeightBold!  
 160 WeightExtraBold!  
 176 WeightHeavy!  
 192 WeightExtraHeavy!  
 208 WeightBlack!  
 224 WeightExtraBlack!  
 240 WeightUltraHeavy!  
 248 WeightUltraBlack!  
 254 WeightMaximum!

*Width*

Numeric Equivalent or Enumerated Type: Specifies a font width.

0 WidthUnknown!  
 16 WidthUltraCondensed!  
 32 WidthUltraCompressed!  
 48 WidthCompressed!  
 64 WidthExtraCondensed!  
 80 WidthCondensed!  
 96 WidthSemiCondensed!  
 112 WidthNormal!  
 128 WidthSemiExpanded!  
 144 WidthExpanded!  
 160 WidthExtraExpanded!  
 176 WidthUltraExpanded!  
 192 WidthDoubleWide!  
 224 WidthTripleWide!

*Source*

Numeric Equivalent or Enumerated Type: Specifies whether the font source is a .PRS file or a .DRS file.

16 PRSFile!  
 20 DRSFile!

*Type*

Numeric Equivalent or Enumerated Type: Specifies a font type.

134 Speedo!  
 135 Intellifont!  
 138 Type1!  
 139 TrueType!



## *CharacterSet*

Numeric Equivalent or Enumerated Type: Specifies a character set.

- 0 FontMatchASCII!
- 1 FontMatchMultiNational1!
- 2 FontMatchMultiNational2!
- 3 FontMatchBoxDrawing!
- 4 FontMatchTypographicSymbols!
- 5 FontMatchIconicSymbols!
- 6 FontMatchMath!
- 7 FontMatchMathExtension!
- 8 FontMatchGreek!
- 9 FontMatchHebrew!
- 10 FontMatchCyrillic!
- 11 FontMatchJapaneseKana!
- 12 FontMatchUserDefined!
- 13 FontMatchArabic!
- 14 FontMatchArabicScript!

---

|                 |                                                                                                                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Document, Initial Font, select font.                                                                                                                                                                                                                                                      |
| <b>Example</b>  | To specify an initial font, the command is:<br><br><code>DocInitialFont (Name: "Courier New Bold"; PointSize: 12p; Family: FamilyCourier!; Attributes: FontMatchNormal!; Weight: WeightExtraBold!; Width: WidthUnknown!; Source: DRSFile!; Type: TrueType!; CharacterSet: FontMatchASCII!)</code> |
| <b>See Also</b> | ♦ DocInitialFontDlg                                                                                                                                                                                                                                                                               |

---

## **DocInitialFontDlg**

---

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>Purpose</b>  | DocInitialFontDlg displays the Document Initial Font dialog box. |
| <b>Syntax</b>   | DocInitialFontDlg ()                                             |
| <b>Route</b>    | Layout, Document, Initial Font.                                  |
| <b>See Also</b> | ♦ DocInitialFont                                                 |

---

---

## DocMaximize

|                 |                                                                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | DocMaximize fills the screen with the current document.<br>IMPORTANT: If the document is already maximized, macro execution is terminated or redirected to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | DocMaximize ()                                                                                                                                                                             |
| <b>Route</b>    | Choose Maximize from the control menu.                                                                                                                                                     |
| <b>See Also</b> | ♦ AppMaximize ♦ DocMinimize                                                                                                                                                                |

---

## DocMinimize

|                 |                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | DocMinimize reduces the current document to an icon.<br>IMPORTANT: If the document is already minimized, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | DocMinimize ()                                                                                                                                                                          |
| <b>Route</b>    | Choose Minimize from the control menu on the title bar.                                                                                                                                 |
| <b>See Also</b> | ♦ AppMinimize ♦ DocMaximize                                                                                                                                                             |

---

## DocMove

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| <b>Purpose</b>  | DocMove repositions the current document.           |
| <b>Syntax</b>   | DocMove ()                                          |
| <b>Route</b>    | Choose Move from the control menu on the title bar. |
| <b>See Also</b> | ♦ AppMove                                           |

---

## DocNext

|                 |                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | DocNext cycles the current document to the back and the most recently opened document to the front. |
| <b>Syntax</b>   | DocNext ()                                                                                          |
| <b>Route</b>    | Choose Next from the control menu on the title bar.                                                 |
| <b>See Also</b> | ♦ DocPrevious                                                                                       |

---

## DocPrevious

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>Purpose</b>  | DocPrevious opens the document that was opened before the current document. |
| <b>Syntax</b>   | DocPrevious ()                                                              |
| <b>Route</b>    | Choose Previous from the control menu on the title bar.                     |
| <b>See Also</b> | ♦ DocNext                                                                   |

---

## DocRedlineMethodDlg

|                |                                                      |
|----------------|------------------------------------------------------|
| <b>Purpose</b> | DocRedlineMethodDlg displays the Redline dialog box. |
| <b>Syntax</b>  | DocRedlineMethodDlg ()                               |
| <b>Route</b>   | Layout, Document, Redline Method.                    |

---

## DocRestore

|                |                                                                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DocRestore restores the current document to its size before it was minimized or maximized.<br><br>IMPORTANT: If the document is already restored, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                 |                                                        |
|-----------------|--------------------------------------------------------|
| <b>Syntax</b>   | DocRestore ()                                          |
| <b>Route</b>    | Choose Restore from the control menu on the title bar. |
| <b>See Also</b> | ♦ AppRestore ♦ DocMaximize ♦ DocMinimize               |

---

## DocSize

|                |                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DocSize sizes the current document window.<br><br>IMPORTANT: If the current document window is maximized, an error is returned and the macro ends or is redirected to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | DocSize ()                                                                                                                                                                                            |
| <b>Route</b>   | Choose Size from the control menu on the title bar.                                                                                                                                                   |

---

## DocSummaryConfig

|                |                                                                                          |
|----------------|------------------------------------------------------------------------------------------|
| <b>Purpose</b> | DocSummaryConfig specifies fields to include in the current document summary.            |
| <b>Syntax</b>  | DocSummaryConfig (DocSumFieldName: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

### Parameters

|                        |                                                                                  |
|------------------------|----------------------------------------------------------------------------------|
| <i>DocSumFieldName</i> | Numeric Equivalent: Specifies fields to include in the current document summary. |
|                        | 1 Abstract!                                                                      |
|                        | 2 Account!                                                                       |
|                        | 3 Address!                                                                       |
|                        | 4 Attachments!                                                                   |
|                        | 5 Author!                                                                        |
|                        | 6 Authorization!                                                                 |
|                        | 7 BillTo!                                                                        |
|                        | 8 BlindCopy!                                                                     |
|                        | 9 CarbonCopy!                                                                    |
|                        | 10 Category!                                                                     |
|                        | 11 CheckedBy!                                                                    |
|                        | 12 Client!                                                                       |
|                        | 13 Comments!                                                                     |
|                        | 14 CreationDate!                                                                 |
|                        | 15 DateCompleted!                                                                |
|                        | 16 Department!                                                                   |

17 DescriptiveName!  
18 DescriptiveType!  
19 Destination!  
20 Disposition!  
21 Division!  
22 DocumentNumber!  
23 Editor!  
24 ForwardTo!  
25 Group!  
26 Keywords!  
27 Language!  
28 MailStop!  
29 Matter!  
30 Office!  
31 Owner!  
32 Project!  
33 Publisher!  
34 Purpose!  
35 ReceivedFrom!  
36 RecordedBy!  
37 RecordedDate!  
38 Reference!  
39 RevisionDate!  
40 RevisionNotes!  
41 RevisionNumber!  
42 Section!  
43 Security!  
44 Source!  
45 Status!  
46 Subject!  
47 TelephoneNumber!  
48 Typist!  
49 VersionDate!  
50 VersionNotes!  
51 VersionNumber!

---

**Route**

File, Document Summary, Configure button, select options from Available Fields list box.

---

**Example**

To include Author, Subject, and Creation Date fields in the current document summary, the command is:

```
DocSummaryConfig (DocSumFieldName: Author!; DocSumFieldName: Subject!; DocSumFieldName:
CreationDate!)
```

---

**See Also** ♦ DocSumSetFull

---

## DocSummaryDelete

---

**Purpose** DocSummaryDelete deletes the current document's summary information.

---

**Syntax** DocSummaryDelete ()

---

**Route** File, Document Summary, Options, Delete Summary from Document.

---

**See Also** ♦ DocSummaryDlg

---

## DocSummaryDlg

---

**Purpose** DocSummaryDlg displays the Document Summary dialog box.

---

**Syntax** DocSummaryDlg ()

---

**Route** Layout, Document, Summary.

---

**See Also** ♦ DocSummaryDelete

---

## DocSummaryPrint

---

**Purpose** DocSummaryPrint prints the current document summary.

---

**Syntax** DocSummaryPrint ()

---

**Route** File, Document Summary, Options, Print Summary.

---

**See Also** ♦ DocSummaryDlg

# DocSummarySaveAs

**Purpose** DocSummarySaveAs specifies information for the current document summary and saves it under a specified filename.

**Syntax** DocSummarySaveAs(*Filename: Character Expression; ExportType: Numeric Equivalent or Enumerated Type; AppendReplace: Numeric Equivalent or Enumerated Type*)

## Parameters

*Filename* Character Expression: Specifies a filename.

*ExportType* Numeric Equivalent: Specifies the file format for the document summary.

- 1 WordPerfect\_42!
- 2 WordPerfect\_50!
- 3 WordPerfect\_51!
- 4 WordPerfect\_60!
- 30 WordStar\_33!
- 31 WordStar\_331!
- 32 WordStar\_34!
- 33 WordStar\_40!
- 34 WordStar\_50!
- 35 WordStar\_55!
- 36 WordStar\_60!
- 50 WordStar2000\_10!
- 51 WordStar2000\_20!
- 52 WordStar2000\_30!
- 62 MicrosoftWord\_40!
- 63 MicrosoftWord\_50!
- 64 MicrosoftWord\_55!
- 70 WordForWindows\_10!
- 71 WordForWindows\_11!
- 72 WordForWindows\_1A!
- 73 WordForWindows\_20!
- 74 WordForWindows\_20a!
- 75 WordForWindows\_20b!
- 80 RTF!
- 81 FFT!
- 82 DisplayWrite\_40!
- 83 DisplayWrite\_42!
- 84 DisplayWrite\_50!
- 100 MultiMate\_33!
- 101 MultiMateAdvantage\_36!
- 102 MultiMateAdvantageII\_10!
- 103 MultiMate\_40!
- 120 OfficeWriter\_60!
- 121 OfficeWriter\_61!

122 OfficeWriter\_611!  
 123 OfficeWriter\_62!  
 150 RTF!  
 160 ASCIIStandard!  
 161 ANSIText\_Windows!  
 163 DelimitedText!  
 165 ASCIIStripped!  
 173 Kermit\_7bit\_Transfer!  
 175 NavyDIF!  
 190 XyWriteIIPlus\_355!  
 191 XyWriteIIPlus\_356!  
 213 AmiPro\_12A!  
 310 AmiPro\_10!  
 311 AmiPro\_12!  
 313 AmiPro\_12B!  
 315 AmiPro\_30!  
 656 ANSIDelimitedTextWindows!  
 657 ANSISGenericWordProcessor\_Windows!  
 807 Excel\_40!

*AppendReplace*

Numeric Equivalent: Specifies whether to append or replace existing document summary information.

- 0 Append!
- 1 Replace!

---

**Route**

File, Document Summary, Options button, Save Summary As New Document, specify a filename.

---

**Example**

To specify MYDOC as the filename of the current document summary, the command is:

DocSummarySaveAs (Filename: "MYDOC"; ExportType: WordPerfect\_60!; AppendReplace: Replace!)

---

## DocSumSetFull

---

**Purpose**

DocSumSetFull specifies fields for a user-defined document summary and specifies data, a field type, and a description for each field. Repeat all four parameters for each field included in the document summary, and enclose repeating parameters in braces.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.



---

**Syntax**

DocSumSetFull ({DocSumFieldName: *Numeric Equivalent* or *Enumerated Type*,  
DocSumFieldData: *Character Expression*; DocSumFieldType: *Numeric Equivalent* or  
*Enumerated Type*; DocSumFieldDesc: *Character Expression*})

---

**Parameters**

*DocSumFieldName*

Numeric Equivalent or Enumerated Type: Specifies fields to include in a document summary. Use the other parameters to define the field.

- 1 Abstract!
- 2 Account!
- 3 Address!
- 4 Attachments!
- 5 Author!
- 6 Authorization!
- 7 BillTo!
- 8 BlindCopy!
- 9 CarbonCopy!
- 10 Category!
- 11 CheckedBy!
- 12 Client!
- 13 Comments!
- 14 CreationDate!
- 15 DateCompleted!
- 16 Department!
- 17 DescriptiveName!
- 18 DescriptiveType!
- 19 Destination!
- 20 Disposition!
- 21 Division!
- 22 DocumentNumber!
- 23 Editor!
- 24 ForwardTo!
- 25 Group!
- 26 Keywords!
- 27 Language!
- 28 MailStop!
- 29 Matter!
- 30 Office!
- 31 Owner!
- 32 Project!
- 33 Publisher!
- 34 Purpose!
- 35 ReceivedFrom!
- 36 RecordedBy!
- 37 RecordedDate!
- 38 Reference!
- 39 RevisionDate!
- 40 RevisionNotes!

41 RevisionNumber!  
 42 Section!  
 43 Security!  
 44 Source!  
 45 Status!  
 46 Subject!  
 47 TelephoneNumber!  
 48 Typist!  
 49 VersionDates!  
 50 VersionNotes!  
 51 VersionNumber!  
 65535 HighestNumber!

*DocSumFieldData*

Character Expression: Specifies data contained in a document summary field.

*DocSumFieldType*

Numeric Equivalent or Enumerated Type: Specifies a field type.

1 SingleLine!  
 2 MultiLine!  
 4 Date!  
 8 ReadOnly!

*DocSumFieldDesc*

Character Expression: Specifies a field description.

---

## DotLeader

### Purpose

DotLeader specifies a dot leader character, as an ASCII number, and a number of spaces between each character.

### Syntax

DotLeader (Character: *Numeric Expression*; SpaceCount: *Numeric Expression*)

### Parameters

*Character*

Numeric Expression: Specifies a dot leader character, as an ASCII number.

*SpaceCount*

Numeric Expression: Specifies a number of spaces between visible dot leader characters.

### Route

Layout, Line, Tab Set, Dot Leader Character, enter a character, Spaces Between Characters, enter a number.

### Example

To specify a dot leader character, the command is:

DotLeader (Character: 44; SpaceCount: 5)

---

**See Also** ♦ ?DotLeader

---

## DoubleSidedPrint

---

**Purpose** DoubleSidedPrint turns on or off double-sided printing (Duplex), and selects a binding edge.

---

**Syntax** DoubleSidedPrint (Flags: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Flags* Numeric Equivalent or Enumerated Type: Specifies whether to turn on double-sided printing and select a binding edge.

- 0 None!
- 1 LongEdge!
- 2 ShortEdge!

---

**Route** Layout, Page, Binding/Duplex, select Off, From Short Edge, or From Long Edge.

---

**Example** To turn on double-sided printing and select a short edge for binding, the command is:

DoubleSidedPrint (Flags: ShortEdge!)

---

## DraftZoom100

---

**Purpose** DraftZoom100 switches the view to Draft and zooms the display size of screen text and graphics to 100%.

---

**Syntax** DraftZoom100 ()

---

**Route** Click ViewDraft/Zoom to 100% on the Power Bar.

---

**See Also** ♦ DraftZoomMargin

---

## DraftZoomMargin

|                 |                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | DraftZoomMargin switches the view to Draft and zooms the display size of screen text and graphics to reach the margins. |
| <b>Syntax</b>   | DraftZoomMargin ()                                                                                                      |
| <b>Route</b>    | Click View Draft/Zoom to Margins on the Power Bar.                                                                      |
| <b>See Also</b> | ♦ DraftZoom100                                                                                                          |

---

## DragTextBegin

|                 |                                                                                                                                                                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | <p>DragTextBegin is the opening command in a DragTextBegin-DragTextEnd statement used to move selected text. This statement may contain positioning commands such as PosLineUp or PosPageTop.</p> <p>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.</p> |
| <b>Syntax</b>   | DragTextBegin ()                                                                                                                                                                                                                                                                                          |
| <b>See Also</b> | ♦ DragTextEnd                                                                                                                                                                                                                                                                                             |

---

## DragTextEnd

|                 |                                                                                                                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | <p>DragTextEnd is the closing command in a DragTextBegin-DragTextEnd statement used to move selected text. This statement may contain positioning commands such as PosLineEnd or PosPageBottom.</p> <p>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.</p> |
| <b>Syntax</b>   | DragTextEnd()                                                                                                                                                                                                                                                                                               |
| <b>See Also</b> | ♦ DragTextBegin                                                                                                                                                                                                                                                                                             |

---

## EditComment

---

**Purpose** EditComment edits a comment.

---

**Syntax** EditComment ()

---

**Route** Insert, Comment, Edit.

---

## EditCopy

---

**Purpose** EditCopy copies selected text or graphics to the Clipboard, replacing the contents of the Clipboard.

IMPORTANT: If no text or graphic is selected, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR.

---

**Syntax** EditCopy ()

---

**Route** Edit, Copy.

---

---

## EditCut

---

**Purpose** EditCut deletes selected text or graphics and copies it to the Clipboard, replacing the contents of the Clipboard.

IMPORTANT: If no text or graphic is selected, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR.

---

**Syntax** EditCut ()

---

**Route** Edit, Cut.

---

---

## EditPaste

---

**Purpose** EditPaste copies the Clipboard contents to the insertion point in the current document.

---

**Syntax** EditPaste (ClipboardFormat: *Character Expression*; FormatId: *Numeric Expression*; LinkType: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*ClipboardFormat* Character Expression: Specifies a Clipboard contents format.

*FormatId* Numeric Expression: Specifies the identification number assigned by WordPerfect for the selected clipboard contents. This number varies.

*LinkType* Numeric Equivalent or Enumerated Type: Determines the type of data link.

- 1 DDE!
- 2 OLE!

---

**Route** When the Clipboard is not empty, choose Edit, Paste or Paste Special.

---

**Example** To paste a picture, the command is:

EditPaste (ClipboardFormat: "P"; FormatId: 3; LinkType: DDE!)

---

---

## EndCenterOrAlignment

**Purpose** EndCenterOrAlignment inserts an End of Centering/Alignment code at the insertion point.

---

**Syntax** EndCenterOrAlignment ()

---

**Route** Layout, Line, Other Codes, End Centering/Alignment [END Cntr/Align], Insert.

---

---

## EndnoteCreate

**Purpose** EndnoteCreate opens an Endnote window.

IMPORTANT: Subsequent commands insert text and codes. The Close command closes the window.

---

**Syntax** EndnoteCreate ()

---

**Route** Insert, Endnote, Create.

---

---

**See Also** ♦ Close

---

## EndnoteEdit

---

**Purpose** EndnoteEdit opens a specified Endnote window.

---

**Syntax** EndnoteEdit (EndnoteNumber: *Character Expression*)

---

**Parameters**

*EndnoteNumber* Character Expression: Specifies the number of the endnote to edit. This parameter is a character expression because endnotes can also be numbered with Roman numerals or letters.

---

**Route** Insert, Endnote, Edit.

---

**Example** To open the first endnote, the command is:

EndnoteEdit (EndnoteNumber: "1")

---

**See Also** ♦ Close, EndnoteEditDlg

---

## EndnoteEditDlg

---

**Purpose** EndnoteEditDlg displays the Edit Endnote dialog box.

---

**Syntax** EndnoteEditDlg ()

---

**Route** Insert, Endnote, Edit.

---

**See Also** ♦ EndnoteEdit

---

## EndnoteMinimum

---

**Purpose** EndnoteMinimum specifies the amount of endnote text to keep together.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

|                   |                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | EndnoteMinimum (Size: <i>Measurement Expression</i> )                                                                                                                                                                   |
| <b>Parameters</b> |                                                                                                                                                                                                                         |
| <i>Size</i>       | Measurement Expression: Specifies the amount of endnote text to keep together. The default is 0.5".<br><br>IMPORTANT: If the amount specified in the Size parameter is less than 1, a 0 must precede the decimal point. |
| <b>Route</b>      | Insert, Endnote, Options, Amount of Endnote to Keep Together, specify an amount.                                                                                                                                        |
| <b>Example</b>    | To keep at least .75" of endnote text together on each page, the command is:<br><br>EndnoteMinimum (Size: 0.75")                                                                                                        |
| <b>See Also</b>   | ♦ EndnoteOptionsDlg                                                                                                                                                                                                     |

## EndnoteNewNumber

|                      |                                                                                                                |
|----------------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>       | EndnoteNewNumber inserts a code at the insertion point to begin new numbering for all following endnotes.      |
| <b>Syntax</b>        | EndnoteNewNumber ( EndnoteNumber: <i>Character Expression</i> )                                                |
| <b>Parameters</b>    |                                                                                                                |
| <i>EndnoteNumber</i> | Character Expression: Specifies a new starting number or character for endnote numbers.                        |
| <b>Route</b>         | Insert, Endnote, New Number, enter a new number or character.                                                  |
| <b>Example</b>       | To number subsequent endnotes beginning with IV, the command is:<br><br>EndnoteNewNumber (EndnoteNumber: "IV") |

## EndnoteNumberDecrement

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <b>Purpose</b> | EndnoteNumberDecrement decreases the endnote number by one. |
| <b>Syntax</b>  | EndnoteNumberDecrement ()                                   |



---

**Route** Insert, Endnote, New Number, Decrease.

---

**See Also** ♦ EndnoteNewNumber

---

---

## EndnoteNumberDisplay

---

**Purpose** EndnoteNumberDisplay displays current endnote numbering.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** EndnoteNumberDisplay ()

---

**See Also** ♦ EndnoteNewNumber

---

---

## EndnoteNumberIncrement

---

**Purpose** EndnoteNumberIncrement increases the endnote number by one.

---

**Syntax** EndnoteNumberIncrement ()

---

**Route** Insert, Endnote, New Number, Increase.

---

**See Also** ♦ EndnoteNewNumber

---

---

## EndnoteNumberMethod

---

**Purpose** EndnoteNumberMethod specifies an endnote numbering method.

---

**Syntax** EndnoteNumberMethod (NumberingMethod: *Numeric Equivalent* or *Enumerated Type*; Characters: *Character Expression*)

---

**Parameters**

*NumberingMethod* Numeric Equivalent or Enumerated Type: Specifies an endnote numbering method.

- 0 Numbers!
- 1 LowerLetters!
- 2 UpperLetters!
- 3 LowerRoman!

- 0 Numbers!
- 4 UpperRoman!
- 5 Characters!

*Characters*

Character Expression: Specifies a character(s) to use as the endnote numbering method. Used when Characters! is used in the first parameter.

**Route**

Insert, Endnote, Options, select an option from the Method pop-up list.

**Example**

To specify the asterisk (\*) as an endnote numbering method, the command is:

EndnoteNumberMethod (NumberingMethod: Characters!; Characters: "\*\*")

**See Also**

- ◆ EndnoteNewNumber

## EndnoteOptions

**Purpose**

EndnoteOptions specifies endnote options from the insertion point of the current document.

**Syntax**

EndnoteOptions(NumberingMethod: *Numeric Equivalent* or *Enumerated Type*, Characters: *Character Expression*; BetweenSpacing: *Measurement Expression*; MinimumHeight: *Measurement Expression*)

**Parameters**

*NumberingMethod*

Numeric Equivalent or Enumerated Type: Specifies the endnote numbering method.

- 0 Numbers!
- 1 LowerLetters!
- 2 UpperLetters!
- 3 LowerRoman!
- 4 UpperRoman!
- 5 Characters!

*Characters*

Character Expression: Specifies up to five characters when the numbering method is Characters!.

*BetweenSpacing*

Measurement Expression: Specifies the space between endnotes.

*MinimumHeight*

Measurement Expression: Specifies how much of an endnote must stay together on a page.

**Route**

Insert, Endnote, Options, select options.

|                 |                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b>  | To set endnote options, the command is:<br><br>EndnoteOptions (NumberingMethod: Characters!; Characters: "%"; BetweenSpacing: .1"; MinimumHeight: .5") |
| <b>See Also</b> | ♦ EndnoteOptionsDlg                                                                                                                                    |

---

## EndnoteOptionsDlg

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <b>Purpose</b>  | EndnoteOptionsDlg displays the Endnote Options dialog box. |
| <b>Syntax</b>   | EndnoteOptionsDlg ()                                       |
| <b>Route</b>    | Insert, Endnote, Options.                                  |
| <b>See Also</b> | ♦ EndnoteOptions                                           |

---

## EndnotePlacement

|                   |                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | EndnotePlacement marks where to generate an endnote list and specifies whether to reset numbering for subsequent endnotes.<br><br>IMPORTANT: This command applies to all endnotes up to the EndnotePlacement code, or from the last Endnote Placement code. |
| <b>Syntax</b>     | EndnotePlacement(Restart: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                             |
| <b>Parameters</b> |                                                                                                                                                                                                                                                             |
| <i>Restart</i>    | Numeric Equivalent or Enumerated Type: Specifies whether endnotes begin renumbering after the EndnotePlacement code.<br><br>0 No!<br>1 Yes!                                                                                                                 |
| <b>Route</b>      | Insert, Endnote, Placement, select renumbering option.                                                                                                                                                                                                      |
| <b>Example</b>    | To specify endnote placement with renumbering set to No, the command is:<br><br>EndnotePlacement (Restart: No!)                                                                                                                                             |

---

**See Also** ♦ EndnotePlacementDlg

---

## EndnotePlacementDlg

---

**Purpose** EndnotePlacementDlg displays the Endnote Placement dialog box.

---

**Syntax** EndnotePlacementDlg ()

---

**Route** Insert, Endnote, Placement.

---

**See Also** ♦ EndnotePlacement

---

## EndnoteSetNumberDlg

---

**Purpose** EndnoteSetNumberDlg displays the Endnote Number dialog box.

---

**Syntax** EndnoteSetNumberDlg ()

---

**Route** Insert, Endnote, New Number.

---

## EndnotesSpaceBetween

---

**Purpose** EndnotesSpaceBetween specifies the amount of space between endnotes.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** EndnotesSpaceBetween (Width: *Numeric Expression*)

---

**Parameters**

*Width* Numeric Expression: Specifies the distance between endnotes.

---

**Example** To specify 0.2 inches between endnotes, the command is:

EndnotesSpaceBetween (Width: 0.2")

---

**See Also** ♦ EndnoteOptions

---

## EnterKey

---

**Purpose** EnterKey performs the same function as Enter.

---

**Syntax** EnterKey ()

---

**Route** Press Enter.

---

## EnvelopeDlg

---

**Purpose** EnvelopeDlg displays the Envelope dialog box.

---

**Syntax** EnvelopeDlg ()

---

**Route** Layout, Envelope.

---

## EquationBaseFontDlg

---

**Purpose** EquationBaseFontDlg displays the Equation Font dialog box.

---

**Syntax** EquationBaseFontDlg ()

---

**Route** Graphics, Equation, Font, Equation Font.

---

## EquationCreate

---

**Purpose** EquationCreate creates an equation box and opens the Equation Editor.

---

**Syntax** EquationCreate ()

---

**Route** Graphics, Equation.

---

---

**See Also** ♦ Close

---

---

## EquationPalette

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | EquationPalette specifies whether to display the Equation Palette in the Equation Editor. |
| <b>Syntax</b>     | EquationPalette (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )             |
| <b>Parameters</b> |                                                                                           |
| <i>State</i>      | Numeric Equivalent or Enumerated Type:<br>0 Off!<br>1 On!                                 |
| <b>Route</b>      | Graphics, Equation, View, Palette.                                                        |
| <b>Example</b>    | To display the Equation palette, the command is:<br>EquationPalette (State: On!)          |

---

## EquationPaletteToggle

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <b>Purpose</b>  | EquationPaletteToggle toggles the Equation Editor palette. |
| <b>Syntax</b>   | EquationPaletteToggle ()                                   |
| <b>Route</b>    | Graphics, Equation, View, Palette.                         |
| <b>See Also</b> | ♦ EquationPalette                                          |

---

## EquationRetrieveDlg

|                |                                                                     |
|----------------|---------------------------------------------------------------------|
| <b>Purpose</b> | EquationRetrieveDlg displays the Retrieve Equation Text dialog box. |
| <b>Syntax</b>  | EquationRetrieveDlg ()                                              |
| <b>Route</b>   | Graphics, Equation, File, Insert File.                              |

---

## EquationZoomDlg

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | EquationZoomDlg displays the Zoom Equation Display dialog box. |
| <b>Syntax</b>  | EquationZoomDlg ()                                             |
| <b>Route</b>   | Graphics, Equation, View, Zoom Display.                        |

---

## ExpandDlg

|                |                                                           |
|----------------|-----------------------------------------------------------|
| <b>Purpose</b> | ExpandDlg displays the Expand Master Document dialog box. |
| <b>Syntax</b>  | ExpandDlg ()                                              |
| <b>Route</b>   | File, Master Document, Expand Master.                     |

---

## FigurePaletteDlg

|                 |                                                                         |
|-----------------|-------------------------------------------------------------------------|
| <b>Purpose</b>  | FigurePaletteDlg displays the Image Tools palette in the Figure Editor. |
| <b>Syntax</b>   | FigurePaletteDlg ()                                                     |
| <b>Route</b>    | Choose Tools from the Graphics feature bar.                             |
| <b>See Also</b> | ♦ FigurePaletteDlgEnd                                                   |

---

## FigurePaletteDlgEnd

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| <b>Purpose</b>  | FigurePaletteDlgEnd closes the Image Tools palette in the Figure Editor. |
| <b>Syntax</b>   | FigurePaletteDlgEnd ()                                                   |
| <b>Route</b>    | Choose Tools from the Graphics feature bar.                              |
| <b>See Also</b> | ♦ FigurePaletteDlg                                                       |

---

## FigureRetrieveDlg

---

**Purpose** FigureRetrieveDlg displays the Insert Image dialog box.

---

**Syntax** FigureRetrieveDlg ()

---

**Route** Graphics, Figure.

---

---

## FileChangeAttributes

---

**Purpose** FileChangeAttributes changes file attributes.

---

**Syntax** FileChangeAttributes (Filename: *Character Expression*; Attributes: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Filename* Character Expression: Specifies a file to edit.

*Attributes* Numeric Equivalent or Enumerated Type: Specifies file attributes to change.

- 0 Normal!
- 1 ReadOnly!
- 2 Hidden!
- 4 System!
- 32 Archive!

---

**Route** File, Open, File Options, Change Attributes.

---

**Example** To specify Read-Only for a file named TEST, the command is:

FileChangeAttributes (Filename: "C:\WPWIN60\TEST"; Attributes: ReadOnly!)

---

## FileChangeDir

---

**Purpose** FileChangeDir changes the Documents or Graphics Files directory.

---

**Syntax** FileChangeDir (Directory: *Character Expression*)



---

## Parameters

*Directory* Character Expression: Specifies the drive and path (directory) displayed in a dialog box.

---

**Route** Follow the menu path to any dialog box where the directory can be changed. For example, the following routes are valid: File, Open, select a directory; or Graphics, Figure, select a directory.

---

**Example** To specify the Documents directory to C:\WPWIN60\DOCS, the command is:  
FileChangeDir (Directory: "C:\WPWIN60\DOCS")

---

## FileCopy

---

**Purpose** FileCopy copies a source file to a destination file.

---

**Syntax** FileCopy(SourceFile: *Character Expression*; DestinationFile: *Character Expression*)

---

### Parameters

*SourceFile* Character Expression: Specifies the name of a file to copy.

*DestinationFile* Character Expression: Specifies the filename the source file is copied to.

---

**Route** File, Open, File Options, Copy, enter source and destination file names.

---

**Example** To copy a file, the command is:  
FileCopy (SourceFile: "HISDOC"; DestinationFile: "HERDOC")

---

**See Also** ♦ FileDelete ♦ FileMove

---

## FileDelete

---

**Purpose** FileDelete deletes a specified file.

---

**Syntax** FileDelete(Filename: *Character Expression*)

---

## Parameters

|                 |                                                                                          |
|-----------------|------------------------------------------------------------------------------------------|
| <i>Filename</i> | Character Expression: Specifies a file to delete.                                        |
| <b>Route</b>    | File, Open, select file to delete, File Options, Delete, Delete.                         |
| <b>Example</b>  | To delete the file MYDOC, the command is:<br><code>FileDelete (Filename: "MYDOC")</code> |
| <b>See Also</b> | ♦ FileCopy ♦ FileMove                                                                    |

---

## FileInsert

|                      |                                                                                                                                                                                                       |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>       | FileInsert retrieves a file into the current document at the insertion point.                                                                                                                         |
| <b>Syntax</b>        | FileInsert (Filename: <i>Character Expression</i> ; AutoDetect: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; InsertIntoDoc: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )            |
| <b>Parameters</b>    |                                                                                                                                                                                                       |
| <i>Filename</i>      | Character Expression: Specifies the file to insert.                                                                                                                                                   |
| <i>AutoDetect</i>    | Numeric Equivalent or Enumerated Type: Specifies whether Auto Detect is on.<br>0 No!<br>1 Yes!                                                                                                        |
| <i>InsertIntoDoc</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display a prompt before a file is inserted. If no value is specified for this parameter, a prompt is displayed.<br>0 Prompt!<br>1 Insert! |
| <b>Route</b>         | Insert, File, select a file, Insert.                                                                                                                                                                  |
| <b>Example</b>       | To insert a file named DOC1.TXT into the current document, the command is:<br><code>FileInsert (Filename: "DOC1.TXT"; AutoDetect: No!; InsertIntoDoc: Prompt!)</code>                                 |
| <b>See Also</b>      | ♦ FileOpen                                                                                                                                                                                            |

---

## FileInsertDlg

---

**Purpose** FileInsertDlg displays the Insert File dialog box.

---

**Syntax** FileInsertDlg ()

---

**Route** Insert, File.

---

---

## FileMove

---

**Purpose** FileMove renames or moves, or renames and moves a file.

---

**Syntax** FileMove(SourceFile: *Character Expression*; DestinationFile: *Character Expression*)

---

**Parameters**

*SourceFile* Character Expression: Specifies a file to move or rename.

*DestinationFile* Character Expression: Specifies a new filename or location, or new filename and location for the source file.

---

**Route** File, Open, select a file, File Options, Move or Rename, enter path and filename, Move or Rename.

---

**Example** To move and rename the file OLDDOC, the command is:

FileMove (SourceFile: "C:\WPWIN60\OLDDOC"; DestinationFile: "C:\ADATA\NEWDOC")

---

**See Also** ♦ FileDelete ♦ FileCopy

---

---

## FileNew

---

**Purpose** FileNew opens a new document. If the maximum nine documents are open, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR.

---

**Syntax** FileNew ()

---

**Route** File, New.

---

---

# FileOpen

---

## Purpose

FileOpen retrieves a specified file into a new document window.

IMPORTANT: If the maximum nine document windows are open, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR.

---

## Syntax

FileOpen(Filename: *Character Expression*; Format: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

### Filename

Character Expression: Specifies the name of a file to open.

### Format

Numeric Equivalent or Enumerated Type: Specifies the format of a file to open.

- 1 WordPerfect\_42!
- 2 WordPerfect\_50!
- 3 WordPerfect\_51!
- 4 WordPerfect\_60!
- 30 WordStar\_33!
- 31 WordStar\_331!
- 32 WordStar\_34!
- 33 WordStar\_40!
- 34 WordStar\_50!
- 35 WordStar\_55!
- 36 WordStar\_60!
- 50 WordStar2000\_10!
- 51 WordStar2000\_20!
- 52 WordStar2000\_30!
- 62 MicrosoftWord\_40!
- 63 MicrosoftWord\_50!
- 64 MicrosoftWord\_55!
- 70 WordForWindows\_10!
- 71 WordForWindows\_11!
- 72 WordForWindows\_1A!
- 73 WordForWindows\_20!
- 74 WordForWindows\_20a!
- 75 WordForWindows\_20b!
- 80 RTF!
- 81 FFI!
- 82 DisplayWrite\_40!
- 83 DisplayWrite\_42!
- 84 DisplayWrite\_50!
- 100 MultiMate\_33!
- 101 MultiMateAdvantage\_36!
- 102 MultiMateAdvantageII\_10!
- 103 MultiMate\_40!

120 OfficeWriter\_60!  
121 OfficeWriter\_61!  
122 OfficeWriter\_611!  
123 OfficeWriter\_62!  
150 RTF!  
160 ASCIIStandard!  
161 ANSIText\_Windows!  
163 DelimitedText!  
165 ASCIIStripped!  
173 Kermit\_7bit\_Transfer!  
175 NavyDIF!  
190 XyWriteIIIPlus\_355!  
191 XyWriteIIIPlus\_356!  
311 AmiPro\_12!  
312 AmiPro\_12A!  
313 AmiPro\_12B!  
314 AmiPro\_20!  
315 AmiPro\_30!  
656 ANSIDelimitedTextWindows!  
657 ANSISGenericWordProcessor\_Windows!  
807 Excel\_40!

---

**Route** File, Open, enter a filename.

---

**Example** To open the file MYFILE, the command is:  
`FileOpen (Filename: "MYFILE")`

---

**See Also** ♦ Close ♦ FileOpenDlg ♦ LastOpenedAdd

---

---

## FileOpenDlg

---

**Purpose** FileOpenDlg displays the Open File dialog box.

---

**Syntax** FileOpenDlg ()

---

**Route** File, Open.

---

**See Also** ♦ FileOpen

---

---

# FileRemoveDirectory

---

|                   |                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FileRemoveDirectory removes a directory.                                                                  |
| <b>Syntax</b>     | FileRemoveDirectory (Directory: <i>Character Expression</i> )                                             |
| <b>Parameters</b> |                                                                                                           |
| <i>Directory</i>  | Character Expression: Specifies a directory to remove.                                                    |
| <b>Route</b>      | File, Open, select a directory, File Options, Remove Directory, Remove.                                   |
| <b>Example</b>    | To remove a directory named NEW, the command is:<br><br>FileRemoveDirectory (Directory: "C:\WPWIN60\NEW") |

---

# FileSave

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FileSave saves the current file. If the document has not been saved, the Save As dialog box is displayed.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Syntax</b>     | FileSave(Filename: <i>Character Expression</i> ; ExportType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Overwrite: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                                                                                      |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <i>Filename</i>   | Character Expression: Specifies the name of a file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>ExportType</i> | Numeric Equivalent or Enumerated Type: Specifies the file format to save a document in.<br><br><ol style="list-style-type: none"><li>1 WordPerfect_42!</li><li>2 WordPerfect_50!</li><li>3 WordPerfect_51!</li><li>4 WordPerfect_60!</li><li>30 WordStar_33!</li><li>31 WordStar_331!</li><li>32 WordStar_34!</li><li>33 WordStar_40!</li><li>34 WordStar_50!</li><li>35 WordStar_55!</li><li>36 WordStar_60!</li><li>50 WordStar2000_10!</li><li>51 WordStar2000_20!</li><li>52 WordStar2000_30!</li><li>62 MicroSoftWord_40!</li></ol> |

---

63 MicrosoftWord\_50!  
64 MicrosoftWord\_55!  
70 WordForWindows\_10!  
71 WordForWindows\_11!  
72 WordForWindows\_1A!  
73 WordForWindows\_20!  
74 WordForWindows\_20a!  
75 WordForWindows\_20b!  
80 RFT!  
81 FFT!  
82 DisplayWrite\_40!  
83 DisplayWrite\_42!  
84 DisplayWrite\_50!  
100 MultiMate\_33!  
101 MultiMateAdvantage\_36!  
102 MultiMateAdvantageII\_10!  
103 MultiMate\_40!  
120 OfficeWriter\_60!  
121 OfficeWriter\_61!  
122 OfficeWriter\_611!  
123 OfficeWriter\_62!  
150 RTF!  
160 ASCIIStandard!  
161 ANSIText\_Windows!  
163 DelimitedText!  
165 ASCIIStripped!  
173 Kermit\_7bit\_Transfer!  
175 NavyDIF!  
190 XyWriteIIIPlus\_355!  
191 XyWriteIIIPlus\_356!  
311 AmiPro\_12!  
312 AmiPro\_12A!  
313 AmiPro\_12B!  
314 AmiPro\_20!  
315 AmiPro\_30!  
656 ANSIDelimitedTextWindows!  
657 ANSISGeneratedWordProcessor\_Windows!  
807 Excel\_40!

### *Overwrite*

Numeric Equivalent or Enumerated Type: Specifies whether to automatically overwrite a file with the same name, or prompt the user to enter a new filename.

- 0 No!
- 1 Yes!
- 2 Prompt!

|                 |                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | File, Save.                                                                                                                                               |
| <b>Example</b>  | To save the file EXPORT.DOC in WordPerfect 4.2 format, the command is:<br>FileSave (Filename: "EXPORT.DOC"; ExportType: WordPerfect_42!; Overwrite: Yes!) |
| <b>See Also</b> | ♦ FileSaveAsDlg                                                                                                                                           |

## FileSaveAsDlg

|                 |                                                |
|-----------------|------------------------------------------------|
| <b>Purpose</b>  | FileSaveAsDlg displays the Save As dialog box. |
| <b>Syntax</b>   | FileSaveAsDlg ()                               |
| <b>Route</b>    | File, Save As.                                 |
| <b>See Also</b> | ♦ FileSave                                     |

## FillColor

|                |                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FillColor creates a background (or fill) for graphics boxes and tables.<br>IMPORTANT: This command must be preceded by a command which specifies a fill style, such as BoxFill, and followed by a closing command such as BoxUpdateDisplay.                                                                                                                                                                            |
| <b>Syntax</b>  | FillColor (ForeColorName: <i>Character Expression</i> ; ForeRed: <i>Numeric Expression</i> ; ForeGreen: <i>Numeric Expression</i> ; ForeBlue: <i>Numeric Expression</i> ; ForeShade: <i>Numeric Expression</i> ; BackColorName: <i>Character Expression</i> ; BackRed: <i>Numeric Expression</i> ; BackGreen: <i>Numeric Expression</i> ; BackBlue: <i>Numeric Expression</i> ; BackShade: <i>Numeric Expression</i> ) |

### Parameters

|                           |                                                                                                       |
|---------------------------|-------------------------------------------------------------------------------------------------------|
| <i>Fore/BackColorName</i> | Specifies foreground and background colors.                                                           |
| <i>Fore/BackRed</i>       | Numeric Expression: Represents the amount of red in the selected color. Values range from 0 to 255.   |
| <i>Fore/BackGreen</i>     | Numeric Expression: Represents the amount of green in the selected color. Values range from 0 to 255. |



*Fore/BackBlue*

Numeric Expression: Represents the amount of blue in the selected color. Values range from 0 to 255.

*Fore/BackShade*

Numeric Expression: Represents the amount of shading in the selected color. Values range from 0 to 100.

---

**Route**

With a graphic displayed in the current document, choose Graphics, Edit Box, Border/Fill button, select a Fill style, Foreground, select a color, Background, select a color, OK.

---

**Example**

To specify a 40 percent grayscale background for a selected graphic, the command sequence is:

```
BoxEdit (BoxNumber: 1)
BoxFill (FillStyleName: 0)
FillColors (ForeColorName: "Black"; ForeRed: 0; ForeGreen: 0; ForeBlue: 0; ForeShade: 40;
 BackColorName: "White"; BackRed: 255; BackGreen: 255; BackBlue: 255; BackShade: 100)
BoxUpdateDisplay ()
```

---

**See Also**

♦ BoxFill ♦ BoxUpdateDisplay ♦ FillSetWPG2 ♦ FillStyleCreate ♦ FillSetGradient ♦ FillStyleEnd

---

## FillSetGradient

---

**Purpose**

FillSetGradient customizes a gradient fill.

**IMPORTANT:** This command must be preceded by FillStyleCreate or FillStyleEdit and followed by FillStyleEnd.

---

**Syntax**

FillSetGradient (Gradient: *Numeric Equivalent* or *Enumerated Type*; Angle: *Numeric Expression*; XLocation: *Numeric Expression*; YLocation: *Numeric Expression*; Calculate: *Numeric Equivalent* or *Enumerated Type*; State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Gradient*

Numeric Equivalent or Enumerated Type: Determines the type of gradient.

- 0 LinearGradient!
- 1 RadialGradient!
- 2 RectangularGradient!

*Angle*

Numeric Expression: Specifies the rotation angle applied to the style.

*XLocation*

Numeric Expression: Specifies the percentage of horizontal offset. A horizontal offset of 10% places the center of the gradient effect toward the left edge; a horizontal offset of 50% centers the effect horizontally.

|                  |                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>YLocation</i> | Numeric Expression: Specifies the percentage of vertical offset. A vertical offset of 10% places the center of the gradient effect toward the top edge; a horizontal offset of 50% centers the effect vertically.                                                                                                                |
| <i>Calculate</i> | Numeric Equivalent or Enumerated Type: Specifies the number of steps used to create a gradient. The default is 30 steps, meaning the gradient effect is comprised of 30 increments. The maximum number of steps is 999, and the minimum is 1; both produce a solid appearance. The optimum number of steps is approximately 120. |
| <i>State</i>     | Numeric Equivalent or Enumerated Type: Specifies whether the gradient fill is selected. <ul style="list-style-type: none"> <li>0 No!</li> <li>1 Yes!</li> </ul>                                                                                                                                                                  |

---

**Route** Graphics, Graphics Styles, Fill, Create or Edit. Create: enter style name, select Gradient, select option from Gradient pop-up list, select pattern option. Edit: select Gradient, edit selection from the Gradient pop-up list, select pattern option.

---

**Example** To create a gradient fill style named Greenburst with a top left corner offset, using 50 lines, the command sequence is:

```
FillStyleCreate (TemplateName: "Greenburst"; StyleFile: PersonalLibrary!)
FillSetGradient (Gradient: RectangularGradient!; Angle: 0.0; XLocation: 0.1; YLocation: 0.1; Calculate: 50;
State: Yes!)
FillColors (ForeRed: 255; ForeGreen: 255; ForeBlue: 0; ForeShade: 100; BackRed: 0; BackGreen: 127;
BackBlue: 0; BackShade: 100)
FillStyleEnd (CancelSave: Save!)
```

---

**See Also** ♦ FillColors ♦ FillStyleCreate ♦ FillStyleDelete ♦ FillStyleEdit ♦ FillStyleEnd  
♦ ?FillGradientType

---

## FillSetWPG2

---

**Purpose** FillSetWPG2 specifies both a pattern set and a pattern from the set.

---

**Syntax** FillSetWPG2 (PatternSet: *Numeric Expression*; Index: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

*PatternSet* Numeric Expression: Specifies a pattern set. Only pattern set 0 is available.

*Index* Numeric Equivalent or Enumerated Type: Specifies a pattern.

- 0 WPG2\_Solid!
- 1 WPG2\_50Gray!
- 2 WPG2\_25Gray!

3 WPG2\_12Gray!  
4 WPG2\_Crosshatch1!  
5 WPG2\_Crosshatch2!  
6 WPG2\_Crosshatch3!  
7 WPG2\_Crosshatch4!  
8 WPG2\_Lines1!  
9 WPG2\_Lines2!  
10 WPG2\_Lines3!  
11 WPG2\_Lines4!  
12 WPG2\_Lines5!  
13 WPG2\_Lines6!  
14 WPG2\_Lines7!  
15 WPG2\_Lines8!  
16 WPG2\_Checkboard1!  
17 WPG2\_Checkboard2!  
18 WPG2\_Bricks1!  
19 WPG2\_Bricks2!  
20 WPG2\_Fishscale!  
21 WPG2\_Honeycomb!  
22 WPG2\_Patio!  
23 WPG2\_Chainlink!  
24 WPG2\_Weave!  
25 WPG2\_Plaid!  
26 WPG2\_Balls!  
27 WPG2\_Mesh!  
28 WPG2\_Crosses!  
29 WPG2\_Triangles!  
30 WPG2\_Waves!  
31 WPG2\_Arch!

---

**Route** Graphics, Graphics Styles, Fill, Create or Edit. Create: enter style name, Pattern, select pattern option. Edit: Pattern, edit pattern option.

---

**Example** To create a fill style named BrickWork, using pattern set 0 and the index pattern WPG2\_Bricks2!, the command sequence is:

```
FillStyleCreate (TemplateName: "BrickWork"; StyleFile: PersonalLibrary!)
FillSetWPG2 (PatternSet: 0; Index: WPG2_Bricks2!)
FillColors (ForeRed: 192; ForeGreen: 192; ForeBlue: 192; ForeShade: 100; BackRed: 127; BackGreen: 0;
BackBlue: 0; BackShade: 100)
FillStyleEnd (CancelSave: Save!)
```

---

**See Also** ♦ FillStyleCreate ♦ FillStyleEdit ♦ FillColors ♦ ?FillWPG2Pattern

---

# FillStyleCopy

---

**Purpose** FillStyleCopy copies a fill style.

---

**Syntax** FillStyleCopy (StyleName: *Numeric Equivalent* or *Enumerated Type*, FromLibrary: *Numeric Equivalent* or *Enumerated Type*, ToLibrary: *Numeric Equivalent* or *Enumerated Type*, DestName: *Character Expression*)

---

## Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>StyleName</i>   | Numeric Equivalent or Enumerated Type: Specifies a fill style to copy.<br><ul style="list-style-type: none"><li>0 Fill10!</li><li>1 Fill20!</li><li>2 Fill30!</li><li>3 Fill40!</li><li>4 Fill50!</li><li>5 Fill50!</li><li>6 Fill70!</li><li>7 Fill80!</li><li>8 Fill90!</li><li>9 Fill100!</li><li>10 FillButton!</li><li>127 NoFill!</li></ul> |
| <i>FromLibrary</i> | Numeric Equivalent or Enumerated Type: Specifies the location of a fill style to copy.<br><ul style="list-style-type: none"><li>0 CurrentDocument!</li><li>1 PersonalLibrary!</li><li>2 SharedLibrary!</li></ul>                                                                                                                                  |
| <i>ToLibrary</i>   | Numeric Equivalent or Enumerated Type: Specifies the destination of a copied fill style.<br><ul style="list-style-type: none"><li>0 CurrentDocument!</li><li>1 PersonalLibrary!</li><li>2 SharedLibrary!</li></ul>                                                                                                                                |
| <i>DestName</i>    | Character Expression: Renames a copied fill style if a style of the same name already exists in the target location.                                                                                                                                                                                                                              |

---

**Route** Graphics, Graphic Styles, Fill, select a style, Options, Copy, enter a new style name.

---

**Example** To copy a fill style named 10% Fill from the current document to the Personal Library, and to rename the copied style My Style, the command is:

```
FillStyleCopy (StyleName: Fill10!; FromLibrary: CurrentDocument!; ToLibrary: PersonalLibrary!;
DestName: "My Style")
```

To copy a fill style named 50% Fill from the current document to the Personal Library, the command is:

```
FillStyleCopy (StyleName: Fill50!; FromLibrary: CurrentDocument!; ToLibrary: PersonalLibrary!)
```

---

**See Also** ♦ FillStyleCreate ♦ FillStyleDelete ♦ FillStyleEdit,

---

## FillStyleCreate

---

**Purpose** FillStyleCreate creates a new fill style.

---

**Syntax** FillStyleCreate (TemplateName: *Character Expression*; Library: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

*TemplateName* Character Expression: Specifies the name of a new style.

*Library* Numeric Equivalent or Enumerated Type: Specifies the destination of a fill style to create.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Graphics, Graphic Styles, Fill, Create, enter a new style.

---

**Example** To create a style named Button Fill, the command is:

```
FillStyleCreate (TemplateName: "Button Fill"; Library: CurrentDoc!)
FillStyleEnd (CancelSave: Save!)
```

---

**See Also** ♦ FillColors ♦ FillSetGradient ♦ FillSetWPG2 ♦ FillStyleCopy ♦ FillStyleDelete  
♦ FillStyleEdit ♦ FillStyleEnd

---

## FillStyleDelete

---

**Purpose** FillStyleDelete deletes a style.

---

**Syntax** FillStyleDelete (StyleName: *Character Expression*; Library: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

|                  |                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <i>StyleName</i> | Character Expression: Specifies the name of a style to delete.                                                                     |
| <i>Library</i>   | Numeric Equivalent or Enumerated Type: Specifies the source of a style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary! |

---

**Route** Graphics, Graphic Styles, Fill, select a style, Options button, Delete.

---

**Example** To delete a style named My Style, the command is:  
FillStyleDelete (StyleName: "My Style"; Library: CurrentDoc!)

---

**See Also** ♦ FillStyleCopy ♦ FillStyleCreate ♦ FillStyleEdit

---

## FillStyleEdit

---

**Purpose** FillStyleEdit is the opening command of a FillStyleEdit-FillStyleEnd statement; this statement contains style commands.

---

**Syntax** FillStyleEdit (Style: *Numeric Equivalent* or *Enumerated Type* or *Character Expression*;  
Library: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

|              |                                                                                                                                                                                                                                                                             |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i> | Numeric Equivalent or Enumerated Type or Character Expression: Specifies the name of the style to edit.<br>0 Fill10!<br>1 Fill20!<br>2 Fill30!<br>3 Fill40!<br>4 Fill50!<br>5 Fill60!<br>6 Fill70!<br>7 Fill80!<br>8 Fill90!<br>9 Fill100!<br>10 FillButton!<br>127 NoFill! |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                 |                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| <i>Library</i>  | Numeric Equivalent or Enumerated Type: Specifies the source of a style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary! |
| <b>Route</b>    | Graphics, Graphic Styles, Fill, select a style, Edit.                                                                              |
| <b>Example</b>  | To edit the fill style named Fill40, the command is:<br>FillStyleEdit (Style: Fill40!; Library: CurrentDoc!)                       |
| <b>See Also</b> | ♦ FillStyleCreate ♦ FillStyleEnd ♦ ?Fill                                                                                           |

---

## FillStyleEnd

|                   |                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FillStyleEnd saves or cancels changes after creating or editing a style.                                                            |
| <b>Syntax</b>     | FillStyleEnd (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                          |
| <b>Parameters</b> |                                                                                                                                     |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to save changes after creating or editing a style.<br>0 Cancel!<br>1 Save! |
| <b>Route</b>      | Choose OK or Cancel from the Edit Fill Style or Create Fill Style dialog box.                                                       |
| <b>Example</b>    | To save changes after creating a new style, the command is:<br>FillStyleEnd (State: Save!)                                          |
| <b>See Also</b>   | ♦ FillStyleCreate ♦ FillStyleEdit                                                                                                   |

---

## FillStyleName

---

**Purpose** FillStyleName names a user-created fill style.  
IMPORTANT: This command must be preceded by FillStyleEdit and followed by FillStyleEnd.

---

**Syntax** FillStyleName (FillName: *Character Expression*)

---

### Parameters

*FillName* Character Expression: Specifies a fill style name.

---

**Route** Graphics, Graphics Styles, Fill, select a fill style, Options, Rename, enter a name.

---

**Example** To name a 10% grayscale fill NEW STYLE, the command sequence is:

```
FillStyleEdit (FillName: Fill10!; Library: PersonalLibrary!)
FillStyleName (FillName: "NEW STYLE")
FillStyleEnd (State: Save!)
```

---

## FillStyleRetrieve

---

**Purpose** FillStyleRetrieve retrieves a fill style.

---

**Syntax** FillStyleRetrieve (Filename: *Character Expression*; Library: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Filename* Character Expression: Specifies a fill style to retrieve.

*Library* Numeric Equivalent or Enumerated Type: Specifies the location of a fill style.

- 0 CurrentDoc!
  - 1 PersonalLibrary!
  - 2 SharedLibrary!
- 

**Route** Graphics, Graphics Styles, Fill, Options, Retrieve, enter a style name.

---

**Example** To retrieve a fill style named My Style from the current document, the command is:

```
FillStyleRetrieve (Filename: "My Style"; Library: CurrentDoc!)
```



---

**See Also**

- ◆ FillStyleSave

---

## FillStyleSave

|                   |                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FillStyleSave saves a fill style.                                                                                                                 |
| <b>Syntax</b>     | FillStyleSave (Filename: <i>Character Expression</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                             |
| <b>Parameters</b> |                                                                                                                                                   |
| <i>Filename</i>   | Character Expression: Specifies a fill style to save.                                                                                             |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies the location of a fill style to save.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary! |
| <b>Route</b>      | Graphics, Graphics Styles, Fill, Options, Save As, enter a new name or a new path and name for the style.                                         |
| <b>Example</b>    | To save a fill style named My Style in the current document, the command is:<br><br>FillStyleSave (Filename: "My Style"; Library: CurrentDoc!)    |
| <b>See Also</b>   | ◆ FillStyleRetrieve                                                                                                                               |

---

## FindDlg

|                |                                            |
|----------------|--------------------------------------------|
| <b>Purpose</b> | FindDlg displays the Find Text dialog box. |
| <b>Syntax</b>  | FindDlg ()                                 |
| <b>Route</b>   | Edit, Find.                                |

---

# FindFiles

---

**Purpose** FindFiles locates specified files.

---

**Syntax** FindFiles: (Action: *Numeric Equivalent* or *Enumerated Type*; SearchWPDocsOnly: *Numeric Equivalent* or *Enumerated Type*; AddMatchesToResultsList: *Numeric Equivalent* or *Enumerated Type*; SearchIn: *Numeric Equivalent* or *Enumerated Type*; IndexName: *Character Expression*; SearchFor: *Character Expression*; FilePattern: *Character Expression*; DocumentSummary: *Character Expression*; SearchResultsTitle: *Character Expression*; ShowResultsDialog: *Numeric Equivalent* or *Enumerated Type*; FromDate: *Character Expression*; ToDate: *Character Expression*)

---

## Parameters

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to open or retrieve a file.

- 1 Open!
- 2 Retrieve!

*SearchWPDocsOnly* Numeric Equivalent or Enumerated Type: Specifies whether to search only for WordPerfect documents.

- 0 No!
- 1 Yes!

*AddMatchesToResultsList* Numeric Equivalent or Enumerated Type: Specifies whether to add matching files to search results.

- 0 No!
- 1 Yes!

*SearchIn* Numeric Equivalent or Enumerated Type: Specifies whether to search a specified directory or the entire disk.

- 0 DirectoryOnly!
- 1 SubTree!
- 2 EntireDisk!
- 4 QuickFinderIndex!

*IndexName* Character Expression: Specifies a search path.

*SearchFor* Character Expression: Specifies a file pattern to search.

*FilePattern* Character Expression: Specifies a template to search.

*DocumentSummary* Character Expression: Specifies a document summary pattern to search.

*SearchResultsTitle* Character Expression: Specifies the search results title.

*ShowResultsDialog* Numeric Equivalent or Enumerated Type: Specifies whether to display the results of the search.

- 0 No!
- 1 Yes!

|                 |                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>FromDate</i> | Character Expression: Specifies the beginning date.                                                                                                                                                                                                                                                                                                          |
| <i>ToDate</i>   | Character Expression: Specifies the ending date.                                                                                                                                                                                                                                                                                                             |
| <b>Route</b>    | File, QuickFinder, specify options, Options, More, enter search specifications and select options, Find.                                                                                                                                                                                                                                                     |
| <b>Example</b>  | To find and open a file, the command is:<br><br>FindFiles (Action: Open!; SearchWPDocsOnly: Yes!; AddMatchesToResultsList: Yes!; SearchIn: DirectoryOnly!; IndexName: "\NO CASE"; SearchFor: ".doc.*"; FilePattern: "standard"; DocumentSummary: "report"; SearchResultsTitle: "results"; ShowResultsDialog: Yes!; FromDate: "01/01/92"; ToDate: "12/01/93") |

---

## FirstLineIndent

|                   |                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FirstLineIndent indents the first line of a new paragraph relative to the left margin.                          |
| <b>Syntax</b>     | FirstLineIndent (Width: <i>Measurement Expression</i> )                                                         |
| <b>Parameters</b> |                                                                                                                 |
| <i>Width</i>      | Measurement Expression: Specifies an amount to indent the first line of a new paragraph.                        |
| <b>Route</b>      | Layout, Paragraph, Format, enter a measurement in the First Line Indent box.                                    |
| <b>Example</b>    | To indent the first line of each new paragraph 1.5 inches, the command is:<br><br>FirstLineIndent (Width: 1.5") |

---

## FloatingCellBegin

|                |                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FloatingCellBegin positions the insertion point at the beginning of the current floating cell when the insertion point is in a floating cell.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | FloatingCellBegin ()                                                                                                                                                                                                                                  |

---

**See Also** ♦ FloatingCellEnd

---

## FloatingCellCreate

---

**Purpose** FloatingCellCreate creates a floating cell.

---

**Syntax** FloatingCellCreate ()

---

**Route** Table, Create, Floating Cell.

---

**See Also** ♦ FloatingCellEdit

---

## FloatingCellEdit

---

**Purpose** FloatingCellEdit searches backwards from the insertion point for a floating cell, then moves the insertion point to the first floating cell found. If no floating cell is found, the command searches forward from the insertion point.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** FloatingCellEdit ()

---

**See Also** ♦ FloatingCellCreate

---

## FloatingCellEnd

---

**Purpose** FloatingCellEnd positions the insertion point at the end of the current floating cell when the insertion point is in a floating cell.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** FloatingCellEnd ()

---

**See Also** ♦ FloatingCellBegin

---

# FloatingCellFormula

|                   |                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FloatingCellFormula specifies a formula used in the current floating cell.<br>IMPORTANT: This command must be preceded by FloatingCellCreate or FloatingCellEdit.                                                         |
| <b>Syntax</b>     | FloatingCellFormula (Formula: <i>Character Expression</i> )                                                                                                                                                               |
| <b>Parameters</b> |                                                                                                                                                                                                                           |
| <i>Formula</i>    | Character Expression: Specifies a formula to place in a floating cell.                                                                                                                                                    |
| <b>Route</b>      | Table, Formula Bar, enter a formula.                                                                                                                                                                                      |
| <b>Example</b>    | To create a floating cell containing a formula to add the values of two other floating cells, the command sequence is:<br><br>FloatingCellCreate ()<br>FloatingCellFormula (Formula: "Floating Cell A + Floating Cell B") |
| <b>See Also</b>   | ♦ FloatingCellCreate ♦ FloatingCellEdit                                                                                                                                                                                   |

---

# FloatingCellNumAlignCurrency

|                   |                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FloatingCellNumAlignCurrency specifies whether to align currency symbols.<br>IMPORTANT: This command must be preceded by FloatingCellCreate or FloatingCellEdit. |
| <b>Syntax</b>     | FloatingCellNumAlignCurrency (Align: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                       |
| <b>Parameters</b> |                                                                                                                                                                  |
| <i>Align</i>      | Numeric Equivalent or Enumerated Type: Designates whether to align currency symbols.<br><br>0 No!<br>1 Yes!                                                      |
| <b>Route</b>      | Table, Number Type, Currency, Custom, Align Currency Symbol.                                                                                                     |
| <b>Example</b>    | To create a floating cell that formats numbers as currency and aligns the currency symbols, the command sequence is:                                             |

FloatingCellCreate ()  
FloatingCellNumberFormat (CellNumType: Currency!)  
FloatingCellNumAlignCurrency (Align: Yes!)

---

**See Also**

♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumberFormat

---

## FloatingCellNumberDateFormat

---

**Purpose**

FloatingCellNumberDateFormat specifies a floating cell date format.

---

**Syntax**

FloatingCellNumberDateFormat (Format: *Numeric Expression*)

---

**Parameters***Format*

Numeric Expression: Specifies a date format. Use the number corresponding to one of the following formats:

- 0 DateFormat!
- 1 March 5, 1993
- 2 3/5/93
- 3 Mar 5, 1993
- 4 5 March 1993
- 5 Friday, March 5, 1993
- 6 4:18 pm
- 7 05Mar93
- 8 March 5, 1993 (4:18pm)
- 9 1993-03-05
- 10 3/05
- 11 Mar 05
- 12 04:18:53 (hh:mm:ss)

---

**Route**

Table, Number Type, Date/Time, Custom, select a format.

---

**Example**

To create a floating cell and specify the date format mm/dd/yy, the command sequence is:

```
FloatingCellCreate ()
FloatingCellNumberFormat (CellNumType: DateFormat!)
FloatingCellNumberDateFormat (Format: 2)
TableFormulaCBar ()
FloatingCellFormula (Formula: "Date() ")
```

---

**See Also**

♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumberFormat

---

# FloatingCellNumberDecDigits

|                   |                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FloatingCellNumberDecDigits specifies the number of digits to display after the decimal point in a floating cell.                                                                                  |
| <b>Syntax</b>     | FloatingCellNumberDecDigits (Number: <i>Numeric Expression</i> )                                                                                                                                   |
| <b>Parameters</b> |                                                                                                                                                                                                    |
| <i>Number</i>     | Numeric Expression: Specifies the number of digits to display after a decimal point.                                                                                                               |
| <b>Route</b>      | Table, Number Type, Fixed, Custom, specify a number of digits after decimal.                                                                                                                       |
| <b>Example</b>    | To edit a floating cell and display two digits after the decimal point, the command sequence is:<br><br>FloatingCellNumberFormat (CellNumType: Fixed2!)<br>FloatingCellNumberDecDigits (Number: 2) |
| <b>See Also</b>   | • FloatingCellCreate • FloatingCellEdit • FloatingCellNumberFormat                                                                                                                                 |

---

# FloatingCellNumberFormat

|                    |                                                                                                                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | FloatingCellNumberFormat specifies a number type for floating cells.<br><br>IMPORTANT: This command must be preceded by FloatingCellCreate or FloatingCellEdit.                                                                                     |
| <b>Syntax</b>      | FloatingCellNumberFormat (CellNumType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                        |
| <b>Parameters</b>  |                                                                                                                                                                                                                                                     |
| <i>CellNumType</i> | Numeric Equivalent or Enumerated Type: Specifies the format of a number in floating cells.<br><br>0 General!<br>1 Integer!<br>2 Fixed2!<br>3 Percent!<br>4 Currency!<br>5 Accounting!<br>6 Commas!<br>7 Scientific!<br>8 DateFormat!<br>9 TextOnly! |

|                 |                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, select a format from the Available Types group box.                                                                                                       |
| <b>Example</b>  | To create a floating cell in the scientific notation format, the command sequence is:<br><pre>FloatingCellCreate () FloatingCellNumberFormat (CellNumType: Scientific!)</pre> |
| <b>See Also</b> | ♦ FloatingCellCreate ♦ FloatingCellEdit                                                                                                                                       |

---

## FloatingCellNumberNegNumber

|                   |                                                                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FloatingCellNumberNegNumber specifies how to display negative numbers in a floating cell.                                                                                                                           |
| <b>Syntax</b>     | FloatingCellNumberNegNumber (NegDisplay: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                      |
| <b>Parameters</b> |                                                                                                                                                                                                                     |
| <i>NegDisplay</i> | Numeric Equivalent or Enumerated Type: Specifies the display of negative numbers.<br>0 Minus!<br>1 Parentheses!<br>2 CRDR!                                                                                          |
| <b>Route</b>      | Table, Number Type, Integer, Custom, select a negative numbers option.                                                                                                                                              |
| <b>Example</b>    | To edit a floating cell and display negative numbers in parentheses, the command sequence is:<br><pre>FloatingCellNumberFormat (CellNumType: Integer!) FloatingCellNumberNegNumber (NegDisplay: Parentheses!)</pre> |
| <b>See Also</b>   | ♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumberFormat                                                                                                                                                  |

---

## FloatingCellNumberRound

|                |                                                                                                |
|----------------|------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FloatingCellNumberRound specifies whether to round numbers for calculation in a floating cell. |
|----------------|------------------------------------------------------------------------------------------------|



|                   |                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | FloatingCellNumberRound (Round: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                           |
| <b>Parameters</b> |                                                                                                                                                                                 |
| <i>Round</i>      | Numeric Equivalent or Enumerated Type: Designates whether to round numbers to calculate.<br>0 No!<br>1 Yes!                                                                     |
| <b>Route</b>      | Table, Number Type, Fixed, Custom, Round for Calculation.                                                                                                                       |
| <b>Example</b>    | To edit a floating cell and round numbers for calculation, the command sequence is:<br>FloatingCellNumberFormat (CellNumType: Fixed2!)<br>FloatingCellNumberRound (Round: Yes!) |
| <b>See Also</b>   | ♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumberFormat                                                                                                              |

---

## FloatingCellNumberUseCommas

|                   |                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FloatingCellNumberUseCommas specifies whether to display commas in floating cell numbers.<br><br>IMPORTANT: This command must be preceded by FloatingCellCreate or FloatingCellEdit.                                      |
| <b>Syntax</b>     | FloatingCellNumberUseCommas (Commas: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                |
| <b>Parameters</b> |                                                                                                                                                                                                                           |
| <i>Commas</i>     | Numeric Equivalent or Enumerated Type: Designates whether to use commas in numbers.<br>0 No!<br>1 Yes!                                                                                                                    |
| <b>Route</b>      | Table, Number Type, Commas, Custom, Use Commas.                                                                                                                                                                           |
| <b>Example</b>    | To create a floating cell that displays numbers without showing commas, the command sequence is:<br>FloatingCellCreate ()<br>FloatingCellNumberFormat (CellNumType: Commas!)<br>FloatingCellNumberUseCommas (Commas: No!) |

---

**See Also**

♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumberFormat

---

## FloatingCellNumberUseCurrency

---

**Purpose**

FloatingCellNumberUseCurrency specifies whether to use a currency symbol in a floating cell.

IMPORTANT: This command must be preceded by FloatingCellCreate or FloatingCellEdit.

**Syntax**

FloatingCellNumberUseCurrency (Currency: *Numeric Equivalent* or *Enumerated Type*)

**Parameters***Currency*

Numeric Equivalent or Enumerated Type: Specifies whether to use a currency symbol.

0 No!  
1 Yes!

**Route**

Table, Number Type, Currency, Custom, Use Currency Symbol.

**Example**

To create a floating cell that does not display currency symbols, the command sequence is:

```
FloatingCellCreate ()
FloatingCellNumberFormat (CellNumType: Currency!)
FloatingCellNumberUseCurrency (Currency: No!)
```

**See Also**

♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumAlignCurrency  
♦ FloatingCellNumberFormat

---

## FloatingCellNumSelectCurrency

---

**Purpose**

FloatingCellNumSelectCurrency specifies a currency symbol for floating cells.

IMPORTANT: This command must be preceded by FloatingCellCreate or FloatingCellEdit.

**Syntax**

FloatingCellNumSelectCurrency (IntlCurrency: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*IntlCurrency*

Numeric Equivalent or Enumerated Type: Specifies a currency symbol or country where the currency is used.

- 0 DefaultCurrency!
- 1 Dollar!
- 2 Peso!
- 3 FrenchCanada!
- 4 Canada!
- 5 Brazil!
- 6 Denmark!
- 7 Peseta!
- 8 Croatia!
- 9 CzechRepublic!
- 10 SlovakRepublic!
- 11 Finland!
- 12 FinlandIntl!
- 13 France!
- 14 Germany!
- 15 Greece!
- 16 Hungary!
- 17 Iceland!
- 18 Italy!
- 19 Japan!
- 20 Netherlands!
- 21 Norway!
- 22 NorwayIntl!
- 23 Ruble!
- 24 RussianRuble!
- 25 SouthAfrica!
- 26 Sweden!
- 27 Turkey!
- 28 UnitedKingdom!
- 29 Switzerland!
- 30 Austria!
- 31 Belgium!
- 32 Ecuador!
- 33 ElSalvador!
- 34 Guatemala!
- 35 Honduras!
- 36 Paraguay!
- 37 Venezuela!
- 38 Poland!
- 39 Slovenia!

|                 |                                                                                                                                                                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Currency, Custom, Symbol, select a currency symbol.                                                                                                                                                                   |
| <b>Example</b>  | To edit a cell and specify the dollar sign (\$) as the currency symbol, the command sequence is:<br><br>FloatingCellEdit ()<br>FloatingCellNumberFormat (CellNumType: Currency!)<br>FloatingCellNumSelectCurrency (IntlCurrency: Dollar!) |
| <b>See Also</b> | ♦ FloatingCellCreate ♦ FloatingCellEdit ♦ FloatingCellNumAlignCurrency<br>♦ FloatingCellNumberFormat ♦ FloatingCellNumberUseCurrency                                                                                                      |

---

## FlushRight

|                |                                                  |
|----------------|--------------------------------------------------|
| <b>Purpose</b> | FlushRight aligns text against the right margin. |
| <b>Syntax</b>  | FlushRight ()                                    |
| <b>Route</b>   | Layout, Line, Flush Right.                       |

---

## Font

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | Font specifies a font.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Syntax</b>  | Font (Name: <i>Character Expression</i> ; Family: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Attributes: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Weight: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Width: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Source: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Character Set: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

### Parameters

|               |                                                                                                                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Name</i>   | Character Expression: Specifies a font name.                                                                                                                                                                                                                 |
| <i>Family</i> | Numeric Equivalent or Enumerated Type:<br>256 FamilyITCBenguiat!<br>2304 FamilyTimesRoman!<br>4352 FamilyITCGaramond!<br>6400 FamilyCenturySchoolbook!<br>8448 FamilyBodon!<br>10496 FamilyITCLubalinGraph!<br>12544 FamilyAlbertus!<br>14592 FamilyCourier! |

16640 FamilyHelvetica!  
18688 FamilyITCRonda!  
20736 FamilyMicrostyle!  
22784 FamilyOptima!  
24832 FamilyLetterGothic!  
26880 FamilyBrush!  
28928 FamilyParkAvenue!  
30976 FamilySignetRoundhand!  
33024 FamilyTekton!  
35072 FamilyUncial!  
37120 FamilyOldEnglish!  
39168 FamilyITCZapfDingbats!  
41216 FamilyBroadway!  
43264 FamilyComputer!  
45312 FamilyPTBarnum!  
47360 FamilyPonderosa!  
49408 FamilyMadrone!  
51456 FamilyHobo!  
53504 FamilyRevue!  
55552 FamilyAestheticOrnamented!

*Attributes*

Numeric Equivalent or Enumerated Type:

0 FontMatchNormal!  
1 FontMatchItalic!  
2 FontMatchOutline!  
4 FontMatchShadow!  
8 FontMatchSmallCaps!

*Weight*

Numeric Equivalent or Enumerated Type:

0 WeightUnknown!  
1 WeightUltraThin!  
8 WeightUltraLight!  
16 WeightExtraThin!  
24 WeightThin!  
32 WeightExtraLight!  
48 WeightLight!  
64 WeightDemiLight!  
80 WeightSemiLight!  
88 WeightBook!  
88 WeightRegular!  
104 WeightRoman!  
112 WeightMedium!  
128 WeightDemiBold!  
136 WeightSemiBold!  
144 WeightBold!  
160 WeightExtraBold!  
176 WeightHeavy!  
192 WeightExtraHeavy!

208 WeightBlack!  
224 WeightExtraBlack!  
240 WeightUltraHeavy!  
248 WeightUltraBlack!  
254 WeightMaximum!

*Width*

Numeric Equivalent or Enumerated Type:

0 WidthUnknown!  
16 WidthUltraCondensed!  
32 WidthUltraCompressed!  
48 WidthCompressed!  
64 WidthExtraCondensed!  
80 WidthCondensed!  
96 WidthSemiCondensed!  
112 WidthNormal!  
128 WidthSemiExpanded!  
144 WidthExpanded!  
160 WidthExtraExpanded!  
176 WidthUltraExpanded!  
192 WidthDoubleWide!  
224 WidthTripleWide!

*Source*

Numeric Equivalent or Enumerated Type:

16 PRSFile!  
20 DRSFile!

*Type*

Numeric Equivalent or Enumerated Type:

134 Speedo!  
135 Intellifont!  
138 Type1!  
139 TrueType!

*CharacterSet*

Numeric Equivalent or Enumerated Type:

0 FontMatchASCII!  
1 FontMatchMultinational1!  
2 FontMatchMultinational2!  
3 FontMatchBoxDrawing!  
4 FontMatchTypographicSymbols!  
5 FontMatchIconicSymbols!  
6 FontMatchMath!  
7 FontMatchMathExtension!  
8 FontMatchGreek!  
9 FontMatchHebrew!  
10 FontMatchCyrillic!  
11 FontMatchJapaneseKana!  
12 FontMatchUserDefined!  
13 FontMatchArabic!  
14 FontMatchArabicScript!

---

|              |                              |
|--------------|------------------------------|
| <b>Route</b> | Layout, Font, select a font. |
|--------------|------------------------------|

---

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Example</b> | To select the font named Courier New Bold, the command is: |
|----------------|------------------------------------------------------------|

```
Font (Name: "Courier New Bold"; Family: FamilyCourier!; Attributes: FontMatchNormal!; Weight:
WeightExtraBold!; Width: WidthUnknown!; Source: DRSFile!; Type: TrueType!; CharacterSet:
FontMatchASCII!)
```

---

## FontDlg

---

|                |                                       |
|----------------|---------------------------------------|
| <b>Purpose</b> | FontDlg displays the Font dialog box. |
|----------------|---------------------------------------|

---

|               |            |
|---------------|------------|
| <b>Syntax</b> | FontDlg () |
|---------------|------------|

---

|              |               |
|--------------|---------------|
| <b>Route</b> | Layout, Font. |
|--------------|---------------|

---

|                 |        |
|-----------------|--------|
| <b>See Also</b> | ♦ Font |
|-----------------|--------|

---

---

## FontDoubleUnderlineToggle

---

|                |                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FontDoubleUnderlineToggle toggles Double Underline.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                              |
|---------------|------------------------------|
| <b>Syntax</b> | FontDoubleUnderlineToggle () |
|---------------|------------------------------|

---

|                 |                             |
|-----------------|-----------------------------|
| <b>See Also</b> | ♦ AttributeAppearanceToggle |
|-----------------|-----------------------------|

---

---

## FontExtraLargeToggle

---

|                |                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FontExtraLargeToggle toggles Extra Large.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeRelativeSizeToggle. |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                         |
|---------------|-------------------------|
| <b>Syntax</b> | FontExtraLargeToggle () |
|---------------|-------------------------|

---

---

**See Also** ♦ AttributeRelativeSizeToggle

---

## FontFineToggle

---

**Purpose** FontFineToggle toggles Fine.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeRelativeSizeToggle.

---

**Syntax** FontFineToggle ()

---

**See Also** ♦ AttributeRelativeSizeToggle

---

## FontHiddenToggle

---

**Purpose** FontHiddenToggle toggles Hidden at the insertion point.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** FontHiddenToggle ()

---

**See Also** ♦ AttributeAppearanceToggle

---

## FontItalicToggle

---

**Purpose** FontItalicToggle toggles Italic.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle.

---

**Syntax** FontItalicToggle ()

---

**See Also** ♦ AttributeAppearanceToggle



---

## FontLargeToggle

---

|                 |                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FontLargeToggle toggles Large.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeRelativeSizeToggle. |
| <b>Syntax</b>   | FontLargeToggle ()                                                                                                                                                                            |
| <b>See Also</b> | ♦ AttributeRelativeSizeToggle                                                                                                                                                                 |

---

## FontOutlineToggle

---

|                 |                                                                                                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FontOutlineToggle toggles Outline.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle. |
| <b>Syntax</b>   | FontOutlineToggle ()                                                                                                                                                                            |
| <b>See Also</b> | ♦ AttributeAppearanceToggle                                                                                                                                                                     |

---

## FontRedlineToggle

---

|                 |                                                                                                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FontRedlineToggle toggles Redline.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle. |
| <b>Syntax</b>   | FontRedlineToggle ()                                                                                                                                                                            |
| <b>See Also</b> | ♦ AttributeAppearanceToggle                                                                                                                                                                     |

---

## FontShadowToggle

---

|                |                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FontShadowToggle toggles Shadow.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                             |
|-----------------|-----------------------------|
| <b>Syntax</b>   | FontShadowToggle ()         |
| <b>See Also</b> | ♦ AttributeAppearanceToggle |

---

## FontSize

---

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FontSize specifies the size of a selected font.                                           |
| <b>Syntax</b>     | FontSize (FontSize: <i>Measurement Expression</i> )                                       |
| <b>Parameters</b> |                                                                                           |
| <i>FontSize</i>   | Measurement Expression: Specifies a font size in points.                                  |
| <b>Route</b>      | Layout, Font, specify a size.                                                             |
| <b>Example</b>    | To specify a 14 point font size, the command is:<br><code>FontSize (FontSize: 14p)</code> |
| <b>See Also</b>   | ♦ Font                                                                                    |

---

## FontSmallCapsToggle

---

|                 |                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FontSmallCapsToggle toggles Small Caps.<br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle. |
| <b>Syntax</b>   | FontSmallCapsToggle ()                                                                                                                                                                                      |
| <b>See Also</b> | ♦ AttributeAppearanceToggle                                                                                                                                                                                 |

---

## FontSmallToggle

---

|                |                                                                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | FontSmallToggle toggles Small.<br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeRelativeSizeToggle. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

**Syntax** FontSmallToggle ()

---

**See Also** ♦ AttributeRelativeSizeToggle

---

---

## FontStrikeoutToggle

---

**Purpose** FontStrikeoutToggle toggles Strikeout.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle.

---

**Syntax** FontStrikeoutToggle ()

---

**See Also** ♦ AttributeAppearanceToggle

---

---

## FontSubscriptToggle

---

**Purpose** FontSubscriptToggle toggles Subscript.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributePositionToggle.

---

**Syntax** FontSubscriptToggle ()

---

**See Also** ♦ AttributePositionToggle

---

---

## FontSuperscriptToggle

---

**Purpose** FontSuperscriptToggle toggles Superscript.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributePositionToggle.

---

**Syntax** FontSuperscriptToggle ()

---

**See Also** ♦ AttributePositionToggle

---

---

## FontUnderlineToggle

|                 |                                                                                                                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FontUnderlineToggle toggles Underline.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeAppearanceToggle. |
| <b>Syntax</b>   | FontUnderlineToggle ()                                                                                                                                                                              |
| <b>See Also</b> | ♦ AttributeAppearanceToggle                                                                                                                                                                         |

---

## FontVeryLargeToggle

|                 |                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FontVeryLargeToggle toggles Very Large.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use AttributeRelativeSizeToggle. |
| <b>Syntax</b>   | FontVeryLargeToggle ()                                                                                                                                                                                 |
| <b>See Also</b> | ♦ AttributeRelativeSizeToggle                                                                                                                                                                          |

---

## FooterA

|                   |                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FooterA specifies whether to create, edit, or cancel Footer A.                                                                                     |
| <b>Syntax</b>     | FooterA (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Occurrence: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )           |
| <b>Parameters</b> |                                                                                                                                                    |
| <i>Action</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to create, edit, or cancel Footer A.<br>0 Off!<br>1 Create!<br>2 Edit!                    |
| <i>Occurrence</i> | Numeric Equivalent or Enumerated Type (optional): Specifies where to display footer A.<br>0 NoPages!<br>1 OddPages!<br>2 EvenPages!<br>3 AllPages! |

|                 |                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Header/Footer, Footer A, Create.                                                                                                                                         |
| <b>Example</b>  | To create Footer A, containing the text "This is Footer A," the command sequence is:<br><pre>FooterA (Action: Create!) Type (Text: "This is Footer A") SubstructureExit ()</pre> |
| <b>See Also</b> | ♦ NewHFWOccurrence ♦ ?FooterAOccur ♦ SubstructureExit                                                                                                                            |

---

## FooterB

|                   |                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FooterB specifies whether to create, edit, or cancel Footer B.                                                                                                              |
| <b>Syntax</b>     | FooterB (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Occurrence: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                    |
| <b>Parameters</b> |                                                                                                                                                                             |
| <i>Action</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to create, edit, or cancel Footer B.<br><pre>0 Off! 1 Create! 2 Edir!</pre>                                        |
| <i>Occurrence</i> | Numeric Equivalent or Enumerated Type (optional): Specifies where to display footer B.<br><pre>0 NoPages! 1 OddPages! 2 EvenPages! 3 AllPages!</pre>                        |
| <b>Route</b>      | Layout, Header/Footer, Footer B, Create.                                                                                                                                    |
| <b>Example</b>    | To edit Footer B to contain the text "This is Footer B," the command sequence is:<br><pre>FooterB (Action: Edit!) Type (Text: "This is Footer B") SubstructureExit ()</pre> |
| <b>See Also</b>   | ♦ NewHFWOccurrence ♦ ?FooterBOccur ♦ SubstructureExit                                                                                                                       |

---

## FooterSeparationDistance

|                   |                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FooterSeparationDistance specifies an amount of blank space above footers.                                                                                      |
| <b>Syntax</b>     | FooterSeparationDistance (Distance: <i>Measurement Expression</i> )                                                                                             |
| <b>Parameters</b> |                                                                                                                                                                 |
| <i>Distance</i>   | Measurement Expression: Specifies an amount of blank space above a footer. The default is .167".                                                                |
| <b>Route</b>      | Layout, Header/Footer, Footer A or Footer B, Create or Edit, Distance, specify a measurement.                                                                   |
| <b>Example</b>    | To specify .5" of blank space above Footer A, the command is:<br><br>FooterA (Action: Edit)<br>FooterSeparationDistance (Distance: 0.5")<br>SubstructureExit () |

---

## FootnoteContinuedMessage

|                   |                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | FootnoteContinuedMessage specifies whether to print "(continued...)" when a footnote is split between two pages.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | FootnoteContinuedMessage (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                   |
| <b>Parameters</b> |                                                                                                                                                                                                                          |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to print a continued message when a footnote is split between two pages.<br><br>0 No!<br>1 Yes!                                                                 |
| <b>Example</b>    | To print a continued message, the command is:<br><br>FootnoteContinuedMessage (State: Yes!)                                                                                                                              |
| <b>See Also</b>   | ♦ FootnoteOptionsDlg                                                                                                                                                                                                     |

---

## FootnoteCreate

|                 |                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | FootnoteCreate opens a Footnote window, where subsequent commands insert text and codes. The SubstructureExit command closes the Footnote window and places a reference to the footnote at the insertion point. |
| <b>Syntax</b>   | FootnoteCreate ()                                                                                                                                                                                               |
| <b>Route</b>    | Insert, Footnote, Create.                                                                                                                                                                                       |
| <b>See Also</b> | ♦ SubstructureExit                                                                                                                                                                                              |

---

## FootnoteEdit

|                       |                                                                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>        | FootnoteEdit opens a specified Footnote window. Subsequent commands apply to the Footnote window until SubstructureExit.                                                                          |
| <b>Syntax</b>         | FootnoteEdit (FootnoteNumber: <i>Character Expression</i> )                                                                                                                                       |
| <b>Parameters</b>     |                                                                                                                                                                                                   |
| <i>FootnoteNumber</i> | Character Expression: Specifies the number of a footnote to edit. Note that this parameter is a character expression. Footnotes can be numbered with arabic numerals, roman numerals, or letters. |
| <b>Route</b>          | Insert, Footnote, Edit.                                                                                                                                                                           |
| <b>Example</b>        | To edit footnote number 3, and enter text, the command is:<br><pre>FootnoteEdit (FootnoteNumber: "3")<br/>Type (Text: "The Cremation of Sam McGee")<br/>SubstructureExit ()</pre>                 |
| <b>See Also</b>       | ♦ FootnoteEditDlg ♦ SubstructureExit                                                                                                                                                              |

---

## FootnoteEditDlg

|                |                                                        |
|----------------|--------------------------------------------------------|
| <b>Purpose</b> | FootnoteEditDlg displays the Edit Footnote dialog box. |
|----------------|--------------------------------------------------------|

---

**Syntax** FootnoteEditDlg ()

---

**Route** Insert, Footnote, Edit.

---

**See Also** ♦ FootnoteEdit

---

## FootnoteMinimum

---

**Purpose** FootnoteMinimum specifies the amount of footnote text to keep together on a page.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** FootnoteMinimum (Width: *Measurement Expression*)

---

### Parameters

*Width* Measurement Expression: Specifies the amount of footnote text to keep together.

---

**Example** To keep 0.5" of footnote text together, the command is:

FootnoteMinimum (Width: 0.5")

---

**See Also** ♦ FootnoteOptions

---

## FootnoteNewNumber

---

**Purpose** FootnoteNewNumber increments subsequent footnotes from a specified number.

---

**Syntax** FootnoteNewNumber (FootnoteNumber: *Character Expression*)

---

### Parameters

*FootnoteNumber* Character Expression: Specifies a new footnote number. Note that this parameter is a character expression. Footnotes can be numbered with arabic numerals, roman numerals, or letters.

---

**Route** Insert, Footnote, New Number, specify a number.

---

**Example** To renumber footnotes beginning with "1", the command is:

FootnoteNewNumber (FootnoteNumber: "1")



---

**See Also** ♦ FootnoteNewNumberDlg

---

## FootnoteNewNumberDlg

---

**Purpose** FootnoteNewNumberDlg displays the Footnote Number dialog box.

---

**Syntax** FootnoteNewNumberDlg ()

---

**Route** Insert, Footnote, New Number.

---

## FootnoteNumberDecrement

---

**Purpose** FootnoteNumberDecrement decreases the last footnote number by one.

---

**Syntax** FootnoteNumberDecrement ()

---

**Route** Insert, Footnote, New Number, Decrease.

---

**See Also** ♦ FootnoteNumberIncrement

---

## FootnoteNumberDisplay

---

**Purpose** FootnoteNumberDisplay displays current footnote numbering.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** FootnoteNumberDisplay ()

---

**See Also** ♦ FootnoteNewNumber

---

## FootnoteNumberIncrement

---

**Purpose** FootnoteNumberIncrement increases the last footnote number by one.

|                 |                                         |
|-----------------|-----------------------------------------|
| <b>Syntax</b>   | FootnoteNumberIncrement ()              |
| <b>Route</b>    | Insert, Footnote, New Number, Increase. |
| <b>See Also</b> | ♦ FootnoteNumberDecrement               |

---

## FootnoteOptions

**Purpose** FootnoteOptions specifies footnote options from the insertion point forward.

**Syntax** FootnoteOptions(NumberingMethod: *Numeric Equivalent* or *Enumerated Type*; Characters: *Character Expression*; BetweenSpacing: *Measurement Expression*; MinimumHeight: *Measurement Expression*; Restart: *Numeric Equivalent* or *Enumerated Type*; PrintContinued: *Numeric Equivalent* or *Enumerated Type*; FootnotePosition: *Numeric Equivalent* or *Enumerated Type*; Separator: *Numeric Equivalent* or *Enumerated Type*)

### Parameters

|                        |                                                                                                                                                                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>NumberingMethod</i> | Numeric Equivalent or Enumerated Type: Specifies the footnote numbering method.<br>0 Numbers!<br>1 LowerLetters!<br>2 UpperLetters!<br>3 LowerRoman!<br>4 UpperRoman!<br>5 Characters!                                                               |
| <i>Characters</i>      | Character Expression: Specifies up to five characters. Used when the numbering method is Characters!.                                                                                                                                                |
| <i>BetweenSpacing</i>  | Measurement Expression: Specifies an amount of space between footnotes.                                                                                                                                                                              |
| <i>MinimumHeight</i>   | Measurement Expression: Specifies how much of a footnote must be kept together on a page.                                                                                                                                                            |
| <i>Restart</i>         | Numeric Equivalent or Enumerated Type: Specifies whether the footnote number restarts with 1 on every page.<br>0 Off!<br>1 On!                                                                                                                       |
| <i>PrintContinued</i>  | Numeric Equivalent or Enumerated Type: Specifies whether "(continued)..." is printed on the last footnote line of the first page and "...(continued)" on the first footnote line of the next page, according to Footnote Options.<br>0 No!<br>1 Yes! |

|                         |                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <i>FootnotePosition</i> | Numeric Equivalent or Enumerated Type: Specifies where to display footnotes.<br>0 BottomOfPage!<br>1 AfterText!          |
| <i>Separator</i>        | Numeric Equivalent or Enumerated Type: Specifies a footnote separator.<br>0 NoLine!<br>1 ShortLine!<br>2 MarginToMargin! |

---

**Route** Insert, Footnote, Options, select options.

---

**Example** To specify footnote options, the command is:  
  
FootnoteOptions (NumberingMethod: Characters!; Characters: "Note"; BetweenSpacing: .1"; MinimumHeight: .5"; Restart: On!; PrintContinued: Yes!; FootnotePosition: BottomOfPage!; Separator: ShortLine!)

---

**See Also** ♦ FootnoteOptionsDlg

---

## FootnoteOptionsDlg

---

**Purpose** FootnoteOptionsDlg displays the Footnote Options dialog box.

---

**Syntax** FootnoteOptionsDlg ()

---

**Route** Insert, Footnote, Options.

---

**See Also** ♦ FootnoteOptions

---

## FootnoteSeparatorLine

---

**Purpose** FootnoteSeparatorLine specifies the characteristics of a separator line between text and footnotes.

FootnoteSeparatorLine (Style: *Numeric Equivalent or Enumerated Type*, SpaceAboveLine: *Measurement Expression*; SpaceBelowLine: *Measurement Expression*; Alignment: *Numeric Equivalent or Enumerated Type*; Width: *Measurement Expression*; Position: *Measurement Expression*)

---

## Parameters

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i>          | Numeric Equivalent or Enumerated Type: Specifies the type of separator line.<br><ul style="list-style-type: none"><li>0 SingleLine!</li><li>1 DoubleLine!</li><li>2 DashedLine!</li><li>3 DottedLine!</li><li>4 ThickLine!</li><li>5 ExtraThickLine!</li><li>6 ThinThickLine!</li><li>7 ThickThinLine!</li><li>8 ButtonTopLeftLine!</li><li>9 ButtonBottomRightLine!</li><li>126 TableDefaultLine!</li><li>127 NoLine!</li></ul> |
| <i>SpaceAboveLine</i> | Measurement Expression: Specifies the amount of space between the separator line and the last line of document text.                                                                                                                                                                                                                                                                                                             |
| <i>SpaceBelowLine</i> | Measurement Expression: Specifies the amount of space between the separator line and the first footnote.                                                                                                                                                                                                                                                                                                                         |
| <i>Alignment</i>      | Numeric Equivalent or Enumerated Type: Specifies how the separator line is justified.<br><ul style="list-style-type: none"><li>0 Left!</li><li>1 Full!</li><li>2 Center!</li><li>3 Right!</li><li>4 SetPosition!</li></ul>                                                                                                                                                                                                       |
| <i>Width</i>          | Measurement Expression: Specifies the width of a separator line.                                                                                                                                                                                                                                                                                                                                                                 |
| <i>Position</i>       | Measurement Expression: Specifies the distance of a separator line from the left edge of the page.                                                                                                                                                                                                                                                                                                                               |

---

**Route** Insert, Footnote, Options, Separator, select options.

---

**Example** To create a separator line with 0.5" of space above the line and 0.25" below, 2" from the left edge of the paper, and 3" in length, the command is:

FootnoteSeparatorLine (Style: DoubleLine!; SpaceAboveLine: 0.5"; SpaceBelowLine: 0.25"; Alignment: SetPosition!; Width: 3.0"; Position: 2.0")

---

**See Also** ♦ FootnoteOptionsDlg

---

## FootnoteSpaceBetween

---

**Purpose** FootnoteSpaceBetween specifies the amount of blank space between footnotes.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** FootnoteSpaceBetween (Space: *Measurement Expression*)

**Parameters**

*Space* Measurement Expression: Specifies the amount of blank space between footnotes. The default is .167".

**Example** To specify .25" blank space between footnotes, the command is:

FootnoteSpaceBetween (Space: .25")

**See Also** ♦ FootnoteOptionsDlg

---

---

## ForcePage

---

**Purpose** ForcePage specifies that a certain page always has an odd or even number.

**Syntax** ForcePage (EvenOddNew: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*EvenOddNew* Numeric Equivalent or Enumerated Type: Specifies whether a page number is odd or even.

0 Even!  
1 Odd!  
2 New!

**Route** Layout, Page, Force Page, select Current Page Odd or Current Page Even or New Page.

**Example** To insert a blank page, forcing a page number to be odd, the command is:

ForcePage (EvenOddNew: Odd!)

---

## ForcePageDlg

---

**Purpose** ForcePageDlg displays the Force Page dialog box.

---

**Syntax** ForcePageDlg ()

---

**Route** Layout, Page, Force Page.

---

## FormatMarginsDlg

---

**Purpose** FormatMarginsDlg displays the Margins dialog box.

---

**Syntax** FormatMarginsDlg ()

---

**Route** Layout, Margins.

---

## GenerateDlg

---

**Purpose** GenerateDlg displays the Generate dialog box.

---

**Syntax** GenerateDlg ()

---

**Route** Tools, Generate.

---

## GenerateDoc

---

**Purpose** GenerateDoc generates Endnotes, Indexes, Tables of Contents, Lists, Cross References, Tables of Authorities, and Master Documents.

---

**Syntax** GenerateDoc (SaveSubdocs: *Numeric Equivalent* or *Enumerated Type*, Undo Action: *Numeric Equivalent* or *Numeric Type*, Hypertext: *Numeric Equivalent* or *Enumerated Type*)

---

---

## Parameters

|                    |                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>SaveSubDocs</i> | Numeric Equivalent or Enumerated Type: Specifies whether to save subdocument changes in a master document. Required only to generate in a master document.<br>0 DontSave!<br>1 Save! |
| <i>UndoAction</i>  | Numeric Equivalent or Enumerated Type: Required only to generate in a master document.<br>0 NoUndo!<br>1 Undo!                                                                       |
| <i>Hypertext</i>   | Numeric Equivalent or Enumerated Type:<br>65535 CreateHypertext!                                                                                                                     |

---

**Route** Tools, Generate, Options, select or deselect Save SubDocuments or Build Hypertext Links.

---

**Example** To generate a document, save subdocuments, and build hypertext links, the command is:  
`GenerateDoc (SaveSubDocs: Save!; Hypertext: -1)`

---

## GetAcceleratorText

---

**Purpose** GetAcceleratorText assigns a macro command's accelerator text to a variable. When a command is mapped to a keystroke, the accelerator text represents the keystroke.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** GetAcceleratorText (Variable: *Variable*; Token: *Macro Command*)

---

### Parameters

*Variable* Variable: Is assigned the accelerator text for the command specified by the Token parameter. Any variable name is a valid value for this parameter.

*Token* Macro Command: Specifies a macro command.

---

**Example** To display the accelerator text for RulerBarShow, the command sequence is:

```
GetAcceleratorText (Variable: Ruler; Token: RulerBarShow)
Type (Ruler)
```

## GetData

---

### Purpose

GetData assigns specific information to a variable about the state of WordPerfect or a WordPerfect document. The macro then uses that variable in conjunction with other commands to retrieve information or to determine whether to perform specified actions.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

---

### Syntax

GetData (Variable: *Variable*; Group: *Numeric Equivalent* or *Enumerated Type*; Item: *Numeric Equivalent* or *Enumerated Type*; Template: *Numeric Equivalent* or *Enumerated Type*; ItemNum: *Numeric Expression*)

---

### Parameters

#### *Variable*

Variable: Is assigned the information specified by the other parameters. Any variable name is a valid value for this parameter.

#### *Group*

Numeric Equivalent or Enumerated Type: Specifies the group a variable belongs to. Summary! extracts data from the document summary by order of entry. SummaryTag! extracts data from the document summary by field tab number.

- 0 Font!
- 1 Printer!
- 2 Abbreviation!
- 3 PaperSize!
- 4 BookMark!
- 5 Summary!
- 6 SummaryTag!
- 7 OutlineStyle!
- 8 Styles!
- 9 Table
- 10 TableNames

#### *Item*

Numeric Equivalent or Enumerated Type: Specifies an item type. For example, Count! returns the number of available fields in the document summary. Name!, Data!, Type!, and Tag! return information extracted from the document summary field specified by the ItemNum parameter.

- 0 Count!
- 1 Name!
- 2 Data!
- 3 Width!
- 4 Length!
- 5 Type!
- 6 Tag!



*Template*

Numeric Equivalent or Enumerated Type: Specifies the location of specified items.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

*ItemNum*

Numeric Expression: Specifies the number of the item from which the Item parameter extracts information. For example, ItemNum can be used to specify a particular document summary field.

---

## Example

The following macros illustrate some uses of GetData. The first two show different ways to list the current document summary field names and field contents. These macros are valid only if a document summary has been created for the current document. The first macro attempts to read every predefined summary field by tag number, while the second determines the number of summary fields in the current document summary, then reads them in order. The third macro displays a list of fonts available in the current document.

Example 1:

```
Type (Text: "****SUMMARY TAGS****")
HardReturn ()
MacroStatusPrompt (State: On!; Prompt: "Getting summary fields by tag number...")
ONERROR (Skipit@)
FOR (Count: 1; Count<=51; Count+1)
GetData (Variable: STName; Group: SummaryTag!; Item: Name!; Template: CurrentDoc!; ItemNum:
Count)
GetData (Variable: STData; Group: SummaryTag!; Item: Data!; Template: CurrentDoc!; ItemNum:
Count)
Type (STName+" "+STData)
HardReturn ()
LABEL (Skipit@)
ENDFOR
```

Example 2:

```
Type (Text: "****SUMMARY****")
HardReturn ()
MacroStatusPrompt (State: On!; Prompt: "Getting summary fields by ordinal...")
GetData (Variable: NrSummaryFields; Group: Summary!; Item: Count!; Template: CurrentDoc!)
ONERROR (Skipit@)
FOR (Count: 1; Count<=NrSummaryFields; Count+1)
GetData (Variable: STName; Group: Summary!; Item: Name!; Template: CurrentDoc! ItemNum: Count)
GetData (Variable: STData; Group: Summary!; Item: Data!; Template: CurrentDoc! ItemNum: Count)
Type (STName+" "+STData)
HardReturn ()
LABEL (Skipit@)
ENDFOR
```

Example 3:

```
NumFonts:= 0 FontName:= " " GetData (Variable: NumFonts; Group: Font!; Item: Count!; Template:
CurrentDoc!)
FOR (FontNumber; 1; FontNumber <= NumFonts; FontNumber + 1)
```

```
GetData (Variable: FontName; Group: Font!; Item: Name!; Template: CurrentDoc!; ItemNum:
 FontNumber)
Type (FontName)
HardReturn ()
ENDFOR Type (Text: "Finished")
HardReturn ()
```

---

## GraphicBoxStyleDlg

---

|                |                                                        |
|----------------|--------------------------------------------------------|
| <b>Purpose</b> | GraphicBoxStyleDlg displays the Box Styles dialog box. |
| <b>Syntax</b>  | GraphicBoxStyleDlg ()                                  |
| <b>Route</b>   | Graphics, choose a box or a box style, Style button.   |

---

## GraphicCreateBox

---

|                 |                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | GraphicCreateBox displays the Insert Image dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | GraphicCreateBox ()                                                                                                                                        |
| <b>See Also</b> | ♦ GraphicEditBox                                                                                                                                           |

---

## GraphicCreateChart

---

|                |                                               |
|----------------|-----------------------------------------------|
| <b>Purpose</b> | GraphicCreateChart creates a chart with Draw. |
| <b>Syntax</b>  | GraphicCreateChart ()                         |
| <b>Route</b>   | Graphics, Chart.                              |

---

---

## GraphicCreateTextArt

|                |                                                                           |
|----------------|---------------------------------------------------------------------------|
| <b>Purpose</b> | GraphicCreateTextArt opens a window for modifying the appearance of text. |
| <b>Syntax</b>  | GraphicCreateTextArt ()                                                   |
| <b>Route</b>   | Graphics, TextArt.                                                        |

---

## GraphicCustomMenuDlg

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| <b>Purpose</b> | GraphicCustomMenuDlg displays the Edit Graphics Menu dialog box. |
| <b>Syntax</b>  | GraphicCustomMenuDlg ()                                          |
| <b>Route</b>   | Graphics, Graphics Styles, Menu.                                 |

---

## GraphicsBoxCaptionDlg

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Purpose</b> | GraphicsBoxCaptionDlg displays the Box Caption dialog box. |
| <b>Syntax</b>  | GraphicsBoxCaptionDlg ()                                   |
| <b>Route</b>   | Graphics, choose a box type, Caption.                      |

---

## GraphicsBoxContentDlg

|                |                                                                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | GraphicsBoxContentDlg displays the Box Content dialog box.                                                                                          |
| <b>Syntax</b>  | GraphicsBoxContentDlg ()                                                                                                                            |
| <b>Route</b>   | Graphics, select a box type, Content. If you choose Figure in the second step: select a graphics file from the Insert File dialog box, OK, Content. |

---

## GraphicsBoxFindDlg

|                |                                                      |
|----------------|------------------------------------------------------|
| <b>Purpose</b> | GraphicsBoxFindDlg displays the Edit Box dialog box. |
| <b>Syntax</b>  | GraphicsBoxFindDlg ()                                |
| <b>Route</b>   | Graphics, Edit Box.                                  |

---

## GraphicsBoxSizeDlg

|                |                                                      |
|----------------|------------------------------------------------------|
| <b>Purpose</b> | GraphicsBoxSizeDlg displays the Box Size dialog box. |
| <b>Syntax</b>  | GraphicsBoxSizeDlg ()                                |
| <b>Route</b>   | Graphics, choose a box type, Size.                   |

---

## GraphicsFigureEditAllDlg

|                |                                                                  |
|----------------|------------------------------------------------------------------|
| <b>Purpose</b> | GraphicsFigureEditAllDlg displays the Image Settings dialog box. |
| <b>Syntax</b>  | GraphicsFigureEditAllDlg ()                                      |
| <b>Route</b>   | Graphics, Edit Box, Tools, Image Settings.                       |

---

## GraphicsLineColor

|                |                                                                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | GraphicsLineColor specifies the color of a graphics line.<br>IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.                     |
| <b>Syntax</b>  | GraphicsLineColor (ColorName: <i>Character Expression</i> ; Red: <i>Numeric Expression</i> ; Green: <i>Numeric Expression</i> ; Blue: <i>Numeric Expression</i> ; Shade: <i>Numeric Expression</i> ) |

---

**Parameters**

|                  |                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------|
| <i>ColorName</i> | Character Expression: Specifies a graphics line color.                                        |
| <i>Red</i>       | Numeric Expression: Represents the amount of red in a color. Values range from 0 to 255.      |
| <i>Green</i>     | Numeric Expression: Represents the amount of green in a color. Values range from 0 to 255.    |
| <i>Blue</i>      | Numeric Expression: Represents the amount of blue in a color. Values range from 0 to 255.     |
| <i>Shade</i>     | Numeric Expression: Represents the shading percentage of a color. Values range from 0 to 100. |

---

**Route** Graphics, Edit Line or Custom Line, Line Color, specify a color.

---

**Example** To specify the color of a graphics line, the command is:

```
GraphicsLineCreate ()
GraphicsLineColor (ColorName: "Red"; Red: 255; Green: 0; Blue: 0; Shade: 100)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ♦ GraphicsLineCreate ♦ GraphicsLineEnd ♦ ?GraphicsLineColorBlue
- ♦ ?GraphicsLineColorGreen ♦ ?GraphicsLineColorRed
- ♦ ?GraphicsLineColorShading

---

## GraphicsLineCreate

---

**Purpose** GraphicsLineCreate creates a graphics line.

---

**Syntax** GraphicsLineCreate (LineType: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

|                 |                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <i>LineType</i> | Numeric Equivalent or Enumerated Type: Specifies the type of graphics line to create.<br>0 HorizontalLine!<br>1 VerticalLine! |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|

---

**Route** Graphics, Custom Line, select horizontal or vertical.

---

**Example** To create a horizontal graphics line, the command is:

```
GraphicsLineCreate (LineType: HorizontalLine!)
```

---

**See Also**

♦ GraphicsLineDelete ♦ GraphicsLineEdit ♦ GraphicsLineEnd ♦ HLineCreate  
♦ VLineCreate

---

## GraphicsLineDelete

**Purpose**

GraphicsLineDelete deletes a graphics line.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax**

GraphicsLineDelete (Line: *Numeric Expression*)

**Parameters***Line*

Numeric Expression: Specifies a line to delete.

**Example**

To delete the first graphics line, the command is:

```
GraphicsLineDelete (Line: 1)
```

**See Also**

♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd

---

## GraphicsLineEdit

**Purpose**

GraphicsLineEdit specifies a graphics line to edit.

**Syntax**

GraphicsLineEdit (Line: *Numeric Expression*)

**Parameters***Line*

Numeric Expression: Specifies a line to edit.

**Route**

Select a line, Graphics, Edit Line.

**Example**

To edit the third graphics line, the command is:

```
GraphicsLineEdit (Line: 3)
```

**See Also**

♦ GraphicsLineCreate ♦ GraphicsLineDelete ♦ GraphicsLineEditNext  
♦ GraphicsLineEditPrevious ♦ GraphicsLineEnd

---

## GraphicsLineEditNext

|                   |                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | GraphicsLineEditNext edits the next graphics line.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | GraphicsLineEditNext (LineType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                  |
| <b>Parameters</b> |                                                                                                                                                        |
| <i>LineType</i>   | Numeric Equivalent or Enumerated Type: Specifies a line type to edit.<br>0 HorizontalLine!<br>1 VerticalLine!                                          |
| <b>Example</b>    | To edit the next horizontal line, the command is:<br><br>GraphicsLineEditNext (LineType: HorizontalLine!)                                              |
| <b>See Also</b>   | ♦ GraphicsLineEdit ♦ GraphicsLineEditPrevious ♦ GraphicsLineEnd                                                                                        |

---

## GraphicsLineEditPrevious

|                   |                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | GraphicsLineEditPrevious edits the previous graphics line.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | GraphicsLineEditPrevious (LineType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                      |
| <b>Parameters</b> |                                                                                                                                                                |
| <i>LineType</i>   | Numeric Equivalent or Enumerated Type: Specifies a line type to edit.<br>0 HorizontalLine!<br>1 VerticalLine!                                                  |
| <b>Example</b>    | To edit the previous vertical line, the command is:<br><br>GraphicsLineEditPrevious (LineType: VerticalLine!)                                                  |
| <b>See Also</b>   | ♦ GraphicsLineEdit ♦ GraphicsLineEditNext ♦ GraphicsLineEnd                                                                                                    |

---

# GraphicsLineEnd

**Purpose** GraphicsLineEnd specifies whether to save changes made to a graphics line.

**Syntax** GraphicsLineEnd (State: *Numeric Equivalent* or *Enumerated Type*)

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to save changes.

- 0 Cancel!
- 1 Save!

**Route** Graphics, Custom Line or Edit Line, select options.

**Example** To save changes after creating or editing a graphics line, the command is:  
GraphicsLineEnd (State: Save!)

**See Also** ♦ GraphicsLineCreate ♦ GraphicsLineDelete ♦ GraphicsLineEdit

---

# GraphicsLineHorizontalPosition

**Purpose** GraphicsLineHorizontalPosition positions a graphics line horizontally.  
IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

**Syntax** GraphicsLineHorizontalPosition (Position: *Numeric Equivalent* or *Enumerated Type*, AbsPos: *Measurement Expression*, LeftColNum: *Numeric Expression*)

## Parameters

*Position* Numeric Equivalent or Enumerated Type: Specifies the horizontal location of a graphics line.

- 0 Left!
- 2 Center!
- 3 Right!
- 7 FullAlign!
- 8 Set!
- 9 BetweenColumns!



|                   |                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>AbsPos</i>     | Measurement Expression: Specifies the exact position on a page to place the graphics line. Use this parameter if you specified Set! as the Location parameter.        |
| <i>LeftColNum</i> | Numeric Expression: Specifies which column to place the graphics line to the right of. Use this parameter if you specified BetweenColumns! as the Location parameter. |

---

**Route** Graphics, Custom Line or Edit Line, Horizontal, select a horizontal position option.

---

**Example** To center a graphic line horizontally on a page, the command is:

GraphicsLineHorizontalPosition (Position: Center!)

---

**See Also** ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ ?GraphicsLineHorizontalAlign  
 ♦ ?GraphicsLineHorizontalColumn ♦ ?GraphicsLineHorizontalPosition

---

## GraphicsLineLength

---

**Purpose** GraphicsLineLength specifies the length of graphics lines.

IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

IMPORTANT: To use this command, the horizontal or vertical position of the graphics line may not be set to Full in the Create Graphics Line (or Edit Graphics Line) dialog box.

---

**Syntax** GraphicsLineLength (Length: *Measurement Expression*)

---

### Parameters

*Length* Measurement Expression: Specifies the length of a graphics line.

---

**Route** Graphics, Custom Line or Edit Line, Length, specify a length.

---

**Example** To create a 2.5" graphics line, the command is:

GraphicsLineLength (Length: 2.5")

---

**See Also**

- ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd
- ♦ ?GraphicsLineLength

---

## GraphicsLineSpacing

---

**Purpose**

GraphicsLineSpacing specifies the amount of white space between a graphics line and text.

IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

---

**Syntax**

GraphicsLineSpacing (TopSpace: *Measurement Expression*; BottomSpace: *Measurement Expression*)

---

**Parameters***TopSpace*

Measurement Expression: Specifies the amount of space above a horizontal graphics line or to the left of a vertical graphics line between the line and text.

*BottomSpace*

Measurement Expression: Specifies the amount of space below horizontal graphics line or to the right of a vertical graphics line between the line and text. The default is the amount entered for the AmountAboveLine parameter.

---

**Route**

Graphics, Custom Line or Edit Line, select spacing options.

---

**Example**

To specify .25" space above and below a graphics line, the command is:

```
GraphicsLineSpacing (TopSpace: 0.25"; BottomSpace: 0.25")
```

---

**See Also**

- ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd
- ♦ ?GraphicsLineBottomSpacing ♦ ?GraphicsLineTopSpacing

---

# GraphicsLineStyle

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | GraphicsLineStyle modifies the style of a graphics line.<br><br>IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Syntax</b>     | GraphicsLineStyle (Style: <i>Character Expression</i> or <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>Style</i>      | Character Expression or Numeric Equivalent or Enumerated Type: Specifies a graphics line style. This command may be either a character expression specifying a user-defined graphics line style, or one of the predefined types listed below. Select an enumerated type or its numeric equivalent to specify a pre-defined type.<br><br>0 SingleLine!<br>1 DoubleLine!<br>3 DashedLine!<br>4 DottedLine!<br>5 ThickLine!<br>5 ExtraThickLine!<br>6 ThinThickLine!<br>7 ThickThinLine!<br>8 ButtonTopLeftLine!<br>9 ButtonBottomRightLine!<br>126 TableDefaultLine!<br>127 NoLine! |
| <b>Route</b>      | Graphics, Custom Line or Edit Line, select a line style.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Example</b>    | To create a dotted graphics line, the command is:<br><br>GraphicsLineStyle (Style: DottedLine!)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>See Also</b>   | ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd ♦ ?GraphicsLineStyle                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

---

# GraphicsLineThickness

|                |                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | GraphicsLineThickness specifies the width or thickness of a graphics line.<br><br>IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                   |                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | GraphicsLineThickness (Thickness: <i>Measurement Expression</i> )                                                                 |
| <b>Parameters</b> |                                                                                                                                   |
| <i>Thickness</i>  | Measurement Expression: Specifies the thickness of a graphics line.                                                               |
| <b>Route</b>      | Graphics, Custom Line or Edit Line, select a line thickness from the Thickness pop-up palette or select Use Line Style Thickness. |
| <b>Example</b>    | To make a graphics line .75" thick, the command is:<br>GraphicsLineThickness (Thickness: 0.75")                                   |
| <b>See Also</b>   | ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd<br>♦ ?GraphicsLineThickness                                             |

## GraphicsLineType

|                   |                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | GraphicsLineType specifies whether a graphics line is horizontal or vertical.<br>IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd. |
| <b>Syntax</b>     | GraphicsLineType (LineType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                    |
| <b>Parameters</b> |                                                                                                                                                                                                      |
| <i>LineType</i>   | Numeric Equivalent or Enumerated Type: Specifies whether a graphics line is horizontal or vertical.<br>0 Horizontal!<br>1 Vertical!                                                                  |
| <b>Route</b>      | Graphics, Custom Line or Edit Line, Line Type, Horizontal or Vertical.                                                                                                                               |
| <b>Example</b>    | To create a vertical graphics line, the command is:<br>GraphicsLineType (LineType: Vertical!)                                                                                                        |
| <b>See Also</b>   | ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd<br>♦ ?GraphicsLineType                                                                                                                     |

---

## GraphicsLineUpdateDisplay

---

**Purpose** GraphicsLineUpdateDisplay specifies whether to save changes made to a graphics line.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** GraphicsLineUpdateDisplay (State: *Numeric Equivalent* or *Enumerated Type*)

**Purpose**  
*State* Numeric Equivalent or Enumerated Type: Specifies whether to save changes.

0 Cancel!  
1 Save!

**Example** To save changes to a graphics line, the command is:

GraphicslineUpdateDisplay (State: Save!)

---

## GraphicsLineUseColor

---

**Purpose** GraphicsLineUseColor specifies whether a graphics line uses the predefined line style color.

IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

**Syntax** GraphicsLineUseColor (State: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**  
*State* Numeric Equivalent or Enumerated Type: Specifies whether a graphics line uses a predefined line style color.

0 No!  
1 Yes!

**Route** Graphics, Custom Line or Edit Line, select a line color or Use Line Style Color.

**Example** To format a graphics line with the predefined line style color, the command is:

GraphicsLineUseColor (State: Yes!)

---

**See Also**

- ♦ GraphicsLineColor ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd
- ♦ ?GraphicslineUseColor

---

## GraphicsLineVerticalPosition

---

**Purpose**

GraphicsLineVerticalPosition specifies the vertical position of a graphics line.  
IMPORTANT: This command must be preceded by GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

---

**Syntax**

GraphicsLineVerticalPosition (Position: *Numeric Equivalent* or *Enumerated Type*,  
Where: *Measurement Expression*)

---

**Parameters***Position*

Numeric Equivalent or Enumerated Type: Specifies the vertical location of a graphics line.

- 1 Top!
- 2 Center!
- 4 Bottom!
- 6 Baseline!
- 7 FullAlign!
- 8 Set!

*Where*

Measurement Expression: Specifies the exact position of a graphics line. Use this parameter if you specified Set! as the Location parameter.

---

**Route**

Graphics, Custom Line or Edit Line, specify vertical position.

---

**Example**

To position a vertical line at the top of a page, the command is:

```
GraphicsLineVerticalPosition (Position: Top!)
```

---

**See Also**

- ♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd
- ♦ GraphicsLineHorizontalPosition ♦ ?GraphicsLineVerticalAlign
- ♦ ?GraphicsLineVerticalPosition

---

## GraphicsStylesDlg

---

**Purpose**

GraphicsStylesDlg displays the Graphics Styles dialog box.

---

|               |                            |
|---------------|----------------------------|
| <b>Syntax</b> | GraphicsStylesDlg ()       |
| <b>Route</b>  | Graphics, Graphics Styles. |

---

## HardColumnBreak

---

|                 |                                                                     |
|-----------------|---------------------------------------------------------------------|
| <b>Purpose</b>  | HardColumnBreak inserts a hard column break at the insertion point. |
| <b>Syntax</b>   | HardColumnBreak ()                                                  |
| <b>Route</b>    | Layout, Columns, Column Break.                                      |
| <b>See Also</b> | ♦ ColumnsOff                                                        |

---

## HardPageBreak

---

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| <b>Purpose</b> | HardPageBreak inserts a hard page break at the insertion point. |
| <b>Syntax</b>  | HardPageBreak ()                                                |
| <b>Route</b>   | Insert, Page Break.                                             |

---

## HardPageBreakInsert

---

|                 |                                                                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | HardPageBreakInsert inserts a page break in columns without positioning the insertion point in the next column.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | HardPageBreakInsert ()                                                                                                                                                                                                  |
| <b>See Also</b> | ♦ HardPageBreak                                                                                                                                                                                                         |

---

---

## HardReturn

|                 |                                                      |
|-----------------|------------------------------------------------------|
| <b>Purpose</b>  | HardReturn begins a new line at the insertion point. |
| <b>Syntax</b>   | HardReturn ()                                        |
| <b>Route</b>    | Press Enter.                                         |
| <b>See Also</b> | ♦ HardReturnInsert                                   |

---

## HardReturnInsert

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| <b>Purpose</b>  | HardReturnInsert inserts a hard return in Outline mode. |
| <b>Syntax</b>   | HardReturnInsert ()                                     |
| <b>Route</b>    | Press Shift+Enter                                       |
| <b>See Also</b> | ♦ HardReturn                                            |

---

## HardSpace

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| <b>Purpose</b> | HardSpace inserts a Hard Space [HSpace] at the insertion point. |
| <b>Syntax</b>  | HardSpace ()                                                    |
| <b>Route</b>   | Layout, Line, Other Codes, Hard Space [HSPACE], Insert.         |

---

## HeaderA

|                   |                                                                        |
|-------------------|------------------------------------------------------------------------|
| <b>Purpose</b>    | HeaderA specifies whether to create, edit, or cancel Header A.         |
| <b>Syntax</b>     | HeaderA (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                        |



*Action* Numeric Equivalent or Enumerated Type: Specifies whether to create, edit, or cancel Header A.

- 0 Off!
- 1 Create!
- 2 Edit!

*Occurrence* Numeric Equivalent or Enumerated Type: Specifies where to place Header A.

- 0 NoPages!
- 1 OddPages!
- 2 EvenPages!
- 3 AllPages!

---

**Route** Layout, Header/Footer, Header A, Create or Edit or Discontinue.

---

**Example** To edit Header A, the command is:  
HeaderA (Action: Edit!)

---

**See Also** ♦ ?HeaderAOccur ♦ HeaderB

---

## HeaderB

---

**Purpose** HeaderB specifies whether to create, edit, or cancel Header B.

---

**Syntax** HeaderB (Action: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to create, edit, or cancel Header B.

- 0 Off!
- 1 Create!
- 2 Edit!

*Occurrence* Numeric Equivalent or Enumerated Type: Specifies where to place Header B.

- 0 NoPages!
- 1 OddPages!
- 2 EvenPages!
- 3 AllPages!

---

**Route** Layout, Header/Footer, Header B, Create or Edit or Discontinue.

---

**Example**

To create Header B, the command is:

HeaderB (Action: Create!; Occurrence: AllPages!)

---

**See Also**

• ?HeaderBOccur • HeaderA

---

---

## HeaderFooterDistanceDlg

**Purpose**

HeaderFooterDistanceDlg displays the Distance dialog box.

**Syntax**

HeaderFooterDistanceDlg ()

**Route**

Layout, Header/Footer, Create or Edit, Distance button.

---

---

## HeaderFooterDlg

**Purpose**

HeaderFooterDlg displays the Headers/Footers dialog box.

**Syntax**

HeaderFooterDlg ()

**Route**

Layout, Header/Footer.

---

---

## HeaderFooterPlacement

**Purpose**

HeaderFooterPlacement specifies where to place headers and footers.

**Syntax**

HeaderFooterPlacement (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type: Specifies where to place headers and footers.

- 0 NoPages!
- 1 OddPages!
- 2 EvenPages!
- 3 AllPages!

|                |                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Layout, Header/Footer, select a header or footer, Create or Edit, Placement button, select a placement option. |
| <b>Example</b> | To place headers and footers on all pages, the command is:<br>HeaderFooterPlacement (State: AllPages!)         |

---

## HeaderFooterPlacementDlg

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| <b>Purpose</b>  | HeaderFooterPlacementDlg displays the Placement dialog box. |
| <b>Syntax</b>   | HeaderFooterPlacementDlg ()                                 |
| <b>Route</b>    | Layout, Header/Footer, Create or Edit, Placement button.    |
| <b>See Also</b> | ♦ NewHFWOccurrence                                          |

---

## HeaderSeparationDistance

|                   |                                                                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | HeaderSeparationDistance sets the amount of blank space below headers.                                                                                                                          |
| <b>Syntax</b>     | HeaderSeparationDistance (Distance: <i>Measurement Expression</i> )                                                                                                                             |
| <b>Parameters</b> |                                                                                                                                                                                                 |
| <i>Distance</i>   | Measurement Expression: Specifies the amount of blank space below a header.<br>IMPORTANT: If the amount specified in the Distance parameter is less than 1, a 0 must precede the decimal point. |
| <b>Route</b>      | Layout, Header/Footer, Create or Edit, Distance, enter an amount.                                                                                                                               |
| <b>Example</b>    | To specify .5" blank space below headers, the command is:<br>HeaderSeparationDistance (Distance: 0.5")                                                                                          |
| <b>See Also</b>   | ♦ ?HeaderSeparator                                                                                                                                                                              |

---

## HelpCoachDlg

|                 |                                             |
|-----------------|---------------------------------------------|
| <b>Purpose</b>  | HelpCoachDlg displays the Coach dialog box. |
| <b>Syntax</b>   | HelpCoachDlg ()                             |
| <b>Route</b>    | Help, Coach.                                |
| <b>See Also</b> | ♦ HelpTutorial                              |

---

## HelpContents

|                |                                             |
|----------------|---------------------------------------------|
| <b>Purpose</b> | HelpContents displays the contents of Help. |
| <b>Syntax</b>  | HelpContents ()                             |
| <b>Route</b>   | Help, Contents.                             |

---

## HelpContextSensitive

|                 |                                                                                                                                                |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | HelpContextSensitive displays Context-sensitive Help for an active item. (Use PAUSE to pause a macro when it encounters HelpContextSensitive.) |
| <b>Syntax</b>   | HelpContextSensitive ()                                                                                                                        |
| <b>Route</b>    | Press F1.                                                                                                                                      |
| <b>See Also</b> | ♦ HelpHowDoI ♦ HelpWhatIs                                                                                                                      |

---

## HelpHowDoI

|                 |                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | HelpHowDoI displays a list of features to click for explanations. (Use PAUSE to pause a macro when it encounters HelpHowDoI.) |
| <b>Syntax</b>   | HelpHowDoI ()                                                                                                                 |
| <b>Route</b>    | Help, How Do I.                                                                                                               |
| <b>See Also</b> | ♦ HelpContextSensitive ♦ HelpWhats                                                                                            |

---

## HelpHowToUseHelp

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | HelpHowToUseHelp displays instructions for using Help.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | HelpHowToUseHelp ()                                                                                                                                        |

---

## HelpKeystrokes

|                 |                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | HelpKeystrokes opens Help information about keystrokes and templates.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | HelpKeystrokes ()                                                                                                                                                         |
| <b>See Also</b> | ♦ HelpContents ♦ HelpHowToUseHelp                                                                                                                                         |

---

## HelpMacroManual

|                |                                                      |
|----------------|------------------------------------------------------|
| <b>Purpose</b> | HelpMacroManual opens Help information about macros. |
| <b>Syntax</b>  | HelpMacroManual ()                                   |

---

|                 |                                   |
|-----------------|-----------------------------------|
| <b>Route</b>    | Help, Macros.                     |
| <b>See Also</b> | ♦ HelpContents ♦ HelpHowToUseHelp |

---

## HelpSearchForHelpOn

---

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <b>Purpose</b> | HelpSearchForHelpOn displays the Search dialog box in Help. |
| <b>Syntax</b>  | HelpSearchForHelpOn ()                                      |
| <b>Route</b>   | Help, Search for Help On.                                   |

---

## HelpTutorial

---

|                 |                                       |
|-----------------|---------------------------------------|
| <b>Purpose</b>  | HelpTutorial starts Tutorial in Help. |
| <b>Syntax</b>   | HelpTutorial ()                       |
| <b>Route</b>    | Help, Tutorial.                       |
| <b>See Also</b> | ♦ HelpCoachDlg                        |

---

## HelpWhatIs

---

|                 |                                                                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | HelpWhatIs uses a mouse or keystroke combinations to get information on an active item. (Use PAUSE to pause a macro when it encounters HelpWhatIs.) |
| <b>Syntax</b>   | HelpWhatIs ()                                                                                                                                       |
| <b>Route</b>    | Press Shift+F1.                                                                                                                                     |
| <b>See Also</b> | ♦ HelpContextSensitive ♦ HelpHowDoI                                                                                                                 |

---

---

# HiddenText

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | HiddenText specifies whether the Hidden Text option is on or off for selected text.       |
| <b>Syntax</b>     | HiddenText (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                  |
| <b>Parameters</b> |                                                                                           |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Toggles the Hidden Text option.<br>0 Off!<br>1 On! |
| <b>Route</b>      | Layout, Font, Hidden.                                                                     |
| <b>Example</b>    | To create hidden text, the command is:<br>HiddenText (State: On!)                         |
| <b>See Also</b>   | ♦ HiddenTextShowAll                                                                       |

---

# HiddenTextShowAll

|                   |                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------|
| <b>Purpose</b>    | HiddenTextShowAll shows or hides hidden text.                                           |
| <b>Syntax</b>     | HiddenTextShowAll (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )         |
| <b>Parameters</b> |                                                                                         |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Toggles Show All Hidden Text.<br>0 Off!<br>1 On! |
| <b>Route</b>      | View, Hidden Text.                                                                      |
| <b>Example</b>    | To show hidden text, the command is:<br>HiddenTextShowAll (State: On!)                  |
| <b>See Also</b>   | ♦ HiddenText                                                                            |

---

## HideBars

---

**Purpose** HideBars hides bars such as scroll bars, Button Bars, and ruler bars.

---

**Syntax** HideBars (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether bars are shown or hidden. This command acts as a toggle if the State parameter is omitted.

0 Off!  
1 On!

---

**Route** View, Hide Bars.

---

**Example** To hide bars, the command is:

HideBars (State: On!)

---

## HLineCreate

---

**Purpose** HLineCreate creates a horizontal line in the current document.

---

**Syntax** HLineCreate ()

---

**Route** Graphics, Horizontal Line.

---

## Hypertext

---

**Purpose** Hypertext activates all hypertext links.

---

**Syntax** Hypertext (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether hypertext is active.

0 Inactive!  
1 Active!



|                |                                                                      |
|----------------|----------------------------------------------------------------------|
| <b>Route</b>   | Tools, Hypertext, Activate button.                                   |
| <b>Example</b> | To activate hypertext, the command is:<br>Hypertext (State: Active!) |

---

## HypertextControlBar

|                |                                                         |
|----------------|---------------------------------------------------------|
| <b>Purpose</b> | HypertextControlBar displays the Hypertext control bar. |
| <b>Syntax</b>  | HypertextControlBar ()                                  |
| <b>Route</b>   | Tools, Hypertext.                                       |

---

## HypertextCreate

|                |                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | HypertextCreate identifies a hypertext link type, identifies the location of a bookmark, and specifies whether to identify the hypertext link as highlighted text or as a button.                                          |
| <b>Syntax</b>  | HypertextCreate (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Macro: <i>Character Expression</i> ; Bookmark: <i>Character Expression</i> ; Display: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

### Parameters

|                 |                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Action</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to jump to a bookmark or run a macro.<br>0 GoToBookmark!<br>1 RunMacro!                       |
| <i>Macro</i>    | Character Expression: Specifies the name of a macro or document where a bookmark is located if the RunMacro! style is specified (see Action parameter) |
| <i>Bookmark</i> | Character Expression: Specifies the name of a bookmark within a current document if GoToBookmark is specified for the first parameter.                 |
| <i>Display</i>  | Numeric Equivalent or Enumerated Type: Specifies whether a hypertext link is displayed highlighted or as a button.<br>0 HighlightedText!<br>1 Button!  |

---

|                 |                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Tools, Hypertext, Create button, select options, enter a bookmark, document, or macro name.                                                                                              |
| <b>Example</b>  | To create a hypertext link to a bookmark named TEST and display the link as a button, the command is:<br><br>HypertextCreate (Action: GoToBookmark!; Bookmark: "TEST"; Display: Button!) |
| <b>See Also</b> | ♦ HypertextExecute                                                                                                                                                                       |

---

## HypertextCreateDlg

---

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| <b>Purpose</b> | HypertextCreateDlg displays the Create Hypertext Link dialog box. |
| <b>Syntax</b>  | HypertextCreateDlg ()                                             |
| <b>Route</b>   | Tools, Hypertext, Create button.                                  |

---

## HypertextDelete

---

|                |                                                                          |
|----------------|--------------------------------------------------------------------------|
| <b>Purpose</b> | HypertextDelete deletes a hypertext link containing the insertion point. |
| <b>Syntax</b>  | HypertextDelete ()                                                       |
| <b>Route</b>   | Tools, Hypertext, Delete.                                                |

---

## HypertextEdit

---

|                |                                                                                                                                                                                                                             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | HypertextEdit modifies a hypertext link.                                                                                                                                                                                    |
| <b>Syntax</b>  | HypertextEdit (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Macro: <i>Character Expression</i> ; Bookmark: <i>Character Expression</i> ; Appearance: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

---

**Parameters**

|                   |                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Action</i>     | Numeric Equivalent or Enumerated Type: Specifies whether a hypertext link jumps to a bookmark or runs a macro.<br><br>0 GoToBookmark!<br>1 RunMacro!             |
| <i>Macro</i>      | Character Expression : Specifies a macro to run.                                                                                                                 |
| <i>Bookmark</i>   | Character Expression: Specifies a bookmark to jump to.                                                                                                           |
| <i>Appearance</i> | Numeric Equivalent or Enumerated Type: Specifies whether a hypertext link will appear as highlighted text or as a button.<br><br>0 HighlightedText!<br>1 Button! |

---

**Route** Tools, Hypertext, Edit, select options.

---

**Example** To connect a hypertext link to a bookmark named Important, and to display the link as a button, the command is:  
  
HypertextEdit (Action: GoToBookmark!; Bookmark: "Important"; Appearance: Button!)

---

**See Also** ♦ HypertextCreate ♦ HypertextDelete

---

## HypertextEditDlg

---

**Purpose** HypertextEditDlg displays the Edit Hypertext Link dialog box.

---

**Syntax** HypertextEditDlg ()

---

**Route** Tools, Hypertext, Edit button.

---

## HypertextExecute

---

**Purpose** HypertextExecute performs a hypertext link.

---

**Syntax** HypertextExecute ()

---

**Route** Tools, Hypertext, Perform button.

---

**See Also**

♦ `HypertextCreate` ♦ `HypertextDelete`

---

## **HypertextNext**

---

**Purpose** `HypertextNext` advances to the next hypertext link in a document.

---

**Syntax** `HypertextNext ()`

---

**Route** Tools, Hypertext, Next button.

---

## **HypertextPrevious**

---

**Purpose** `HypertextPrevious` returns to the previous hypertext link in a document.

---

**Syntax** `HypertextPrevious ()`

---

**Route** Tools, Hypertext, Previous button.

---

## **HypertextReturnFrom**

---

**Purpose** `HypertextReturnFrom` returns the insertion point from a bookmark to the bookmark's hypertext link.

---

**Syntax** `HypertextReturnFrom ()`

---

**Route** Tools, Hypertext, Back button.

---

---

# Hyphen

**Purpose** Hyphen inserts a hyphen character at the insertion point.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** Hyphen ()

---

---

# Hyphenation

**Purpose** Hyphenation specifies whether to turn on Hyphenation.

**Syntax** Hyphenation (State: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*State* Numeric Equivalent or Enumerated Type: Specifies whether to turn on Hyphenation.  
0 Off!  
1 On!

**Route** Layout, Line, Hyphenation, select or deselect Hyphenation On.

**Example** To turn on Hyphenation, the command is:  
Hyphenation (State: On!)

**See Also** ♦ HyphenationIgnore ♦ ?Hyphenation

---

---

# HyphenationIgnore

**Purpose** HyphenationIgnore specifies whether to cancel Hyphenation for a particular word.

**Syntax** HyphenationIgnore ()

**Route** Layout, Line, Other Codes, Cancel Hyphenation of Word.

---

---

## HyphenationSoftReturn

|                |                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | HyphenationSoftReturn inserts a Hyphen Soft Return code to indicate where to hyphenate a word if necessary. |
| <b>Syntax</b>  | HyphenationSoftReturn ()                                                                                    |
| <b>Route</b>   | Layout, Line, Other Codes, Hyphenation Soft Return.                                                         |

---

## HyphenationZoneLeft

|                   |                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | HyphenationZoneLeft sets the lower limit of a hyphenation zone. The zone specifies where to hyphenate a word. |
| <b>Syntax</b>     | HyphenationZoneLeft (HZone: <i>Numeric Expression</i> )                                                       |
| <b>Parameters</b> |                                                                                                               |
| <i>HZone</i>      | Numeric Expression: Specifies the lower limit of a hyphenation zone.                                          |
| <b>Route</b>      | Layout, Line, Hyphenation, specify a percentage in the Percent Left entry field.                              |
| <b>Example</b>    | To set the lower limit of a hyphenation zone to 10%, the command is:<br>HyphenationZoneLeft (HZone: 10)       |
| <b>See Also</b>   | ♦ HyphenationZoneRight ♦ ?LeftHyphZone                                                                        |

---

## HyphenationZoneRight

|                   |                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | HyphenationZoneRight sets the upper limit of a hyphenation zone. The zone specifies where to hyphenate a word. |
| <b>Syntax</b>     | HyphenationZoneRight (HZone: <i>Numeric Expression</i> )                                                       |
| <b>Parameters</b> |                                                                                                                |
| <i>HZone</i>      | Numeric Expression: Sets the upper limit of a hyphenation zone.                                                |

|                 |                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Line, Hyphenation Zone, specify a percentage in the Percent Right entry field.                            |
| <b>Example</b>  | To set the upper limit of a hyphenation zone to 4%, the command is:<br><pre>HyphenationZoneRight (HZone: 4)</pre> |
| <b>See Also</b> | ♦ HyphenationZoneLeft ♦ ?RightHyphZone                                                                            |

## ImportCreateDlg

|                |                                                           |
|----------------|-----------------------------------------------------------|
| <b>Purpose</b> | ImportCreateDlg displays the Create Data Link dialog box. |
| <b>Syntax</b>  | ImportCreateDlg ()                                        |
| <b>Route</b>   | Insert, Spreadsheet/Database, Create Link.                |

## ImportDlg

|                |                                                |
|----------------|------------------------------------------------|
| <b>Purpose</b> | ImportDlg displays the Import Data dialog box. |
| <b>Syntax</b>  | ImportDlg ()                                   |
| <b>Route</b>   | Insert, Spreadsheet/Database, Import.          |

## ImportDoEdit

|                 |                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ImportDoEdit modifies and performs an existing link between a spreadsheet or database and the current document. |
| <b>Syntax</b>   | ImportDoEdit ()                                                                                                 |
| <b>Route</b>    | Import, Spreadsheet/Database, Edit Link.                                                                        |
| <b>See Also</b> | ♦ ImportDoImport ♦ ImportDoLink ♦ ImportDoUpdate                                                                |

---

## ImportDoImport

|                 |                                                                                                |
|-----------------|------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ImportDoImport retrieves information from a spreadsheet or database into the current document. |
| <b>Syntax</b>   | ImportDoImport ()                                                                              |
| <b>Route</b>    | Insert, Spreadsheet/Database, Import, select options.                                          |
| <b>See Also</b> | ♦ ImportDoEdit ♦ ImportDoLink ♦ ImportDoUpdate                                                 |

---

## ImportDoLink

|                 |                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ImportDoLink creates and performs a link between a spreadsheet or database and the current document. |
| <b>Syntax</b>   | ImportDoLink ()                                                                                      |
| <b>Route</b>    | Insert, Spreadsheet/Database, Create Link, select options.                                           |
| <b>See Also</b> | ♦ ImportDoEdit ♦ ImportDoImport ♦ ImportDoUpdate                                                     |

---

## ImportDoUpdate

|                 |                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ImportDoUpdate revises linked data in the current document to match the current data in linked spreadsheet or database files. |
| <b>Syntax</b>   | ImportDoUpdate ()                                                                                                             |
| <b>Route</b>    | Insert, Spreadsheet/Database, Update.                                                                                         |
| <b>See Also</b> | ♦ ImportDoEdit ♦ ImportDoImport ♦ ImportDoLink                                                                                |



---

## ImportEditDlg

---

**Purpose** ImportEditDlg displays the Edit Data Link dialog box.

---

**Syntax** ImportEditDlg ()

---

**Route** Insert, Spreadsheet/Database, Edit Link.

---

## ImportOptionsDlg

---

**Purpose** ImportOptionsDlg displays the Link Options dialog box.

---

**Syntax** ImportOptionsDlg ()

---

**Route** Insert, Spreadsheet/Database, Option.

---

---

## ImportSetAsciiEncap

---

**Purpose** ImportSetAsciiEncap specifies an ASCII encapsulation character to use when importing spreadsheet or database data saved as ASCII delimited text.

---

**Syntax** ImportSetAsciiEncap (EncapsulationChar: *Character Expression*)

---

**Parameters**

*EncapsulationChar* Character Expression: Specifies an ASCII encapsulation character for ASCII delimited text.

---

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select ASCII Delimited Text from the Data Type pop-up list, enter encapsulation character.

---

**Example** To use an asterisk (\*) as an encapsulation character, the command is:

ImportSetAsciiEncap (EncapsulationChar: "\*\*")

---

**See Also** ♦ ImportSetAsciiFieldDelimiter ♦ ImportSetAsciiRecordDelimiter  
♦ ImportSetAsciiStrip

---

## ImportSetAsciiFieldDelimiter

---

|                       |                                                                                                                                                                      |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>        | ImportSetAsciiFieldDelimiter specifies characters to mark the beginning or end of a field when importing spreadsheet or database data saved as ASCII delimited text. |
| <b>Syntax</b>         | ImportSetAsciiFieldDelimiter (FieldDelimiter: <i>Character Expression</i> )                                                                                          |
| <b>Parameters</b>     |                                                                                                                                                                      |
| <i>FieldDelimiter</i> | Character Expression: Specifies characters to mark the beginning or end of an ASCII delimited text field.                                                            |
| <b>Route</b>          | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select ASCII Delimited Text from the Data Type pop-up list, enter field delimiters.                |
| <b>Example</b>        | To use a dollar sign (\$) and a percent sign (%) as a field delimiter, the command is:<br><br>ImportSetAsciiFieldDelimiter (FieldDelimiter: "\$%")                   |
| <b>See Also</b>       | ♦ ImportSetAsciiEncap ♦ ImportSetAsciiRecordDelimiter ♦ ImportSetAsciiStrip                                                                                          |

---

## ImportSetAsciiRecordDelimiter

---

|                        |                                                                                                                                                                        |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>         | ImportSetAsciiRecordDelimiter specifies characters to mark the beginning or end of a record when importing spreadsheet or database data saved as ASCII delimited text. |
| <b>Syntax</b>          | ImportSetAsciiRecordDelimiter (RecordDelimiter: <i>Character Expression</i> )                                                                                          |
| <b>Parameters</b>      |                                                                                                                                                                        |
| <i>RecordDelimiter</i> | Character Expression: Specifies characters to mark the beginning or end of an import record.                                                                           |
| <b>Route</b>           | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select ASCII Delimited Text from the Data Type pop-up list, enter record delimiter.                  |
| <b>Example</b>         | To use an ampersand (&) as a record delimiter, the command is:<br><br>ImportSetAsciiRecordDelimiter (RecordDelimiter: "&")                                             |
| <b>See Also</b>        | ♦ ImportSetAsciiEncap ♦ ImportSetAsciiFieldDelimiter ♦ ImportSetAsciiStrip                                                                                             |

---

## ImportSetAsciiStrip

---

|                   |                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetAsciiStrip removes extra characters when importing spreadsheet or database data saved as ASCII delimited text.                                                 |
| <b>Syntax</b>     | ImportSetAsciiStrip (StripChar: <i>Character Expression</i> )                                                                                                           |
| <b>Parameters</b> |                                                                                                                                                                         |
| <i>StripChar</i>  | Character Expression: Specifies characters used to mark text to strip.                                                                                                  |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select ASCII Delimited Text from the Data Type pop-up list, enter character(s) to mark text to strip. |
| <b>Example</b>    | To strip characters marked with an asterisk (*), the command is:<br><br>ImportSetAsciiStrip (StripChar: "**")                                                           |
| <b>See Also</b>   | ♦ ImportSetAsciiEncap ♦ ImportSetAsciiFieldDelimiter<br>♦ ImportSetAsciiRecordDelimiter                                                                                 |

---

## ImportSetDestination

---

|                        |                                                                                                                                                                                       |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>         | ImportSetDestination specifies a format for imported spreadsheet or database data.                                                                                                    |
| <b>Syntax</b>          | ImportSetDestination (DestinationType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                          |
| <b>Parameters</b>      |                                                                                                                                                                                       |
| <i>DestinationType</i> | Numeric Equivalent or Enumerated Type: Specifies a format for imported data.<br><ol style="list-style-type: none"><li>1 WPTable!</li><li>2 TabbedText!</li><li>3 MergeData!</li></ol> |
| <b>Route</b>           | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select an option from the Import As or Link As pop-up list.                                                         |
| <b>Example</b>         | To import data as a table, the command is:<br><br>ImportSetDestination (DestinationType: WPTable!)                                                                                    |

---

**See Also** ♦ ImportSetSource

---

## ImportSetDictPath

---

**Purpose** ImportSetDictPath specifies the NetWare SQL dictionary path for imported spreadsheet or database data.

**Syntax** ImportSetDictPath (DictionaryPath: *Character Expression*)

### Parameters

*DictionaryPath* Character Expression: Specifies a netware SQL dictionary path for imported data.

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select NetWare SQL from the Data Type pop-up list, Logon, enter dictionary path.

**Example** To set the dictionary path c:\wpwin\dict, the command is:

```
ImportSetDictPath (DictionaryPath: "c:\wpwin\dict")
```

**See Also** ♦ ImportSetFilePath ♦ ImportSetPassword ♦ ImportSetUserID

---

## ImportSetFields

---

**Purpose** ImportSetFields specifies database fields to import.

**Syntax** ImportSetFields (FieldNames: *Character Expression*)

### Parameters

*FieldNames* Character Expression: Specifies database fields to import.

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select a database type from the Data Type pop-up list, enter field names.

**Example** To import database “field1, and field2”, the command is:

```
ImportSetFields (FieldNames: "field1", "field2")
```

**See Also** ♦ ImportSetFileName ♦ ImportSetIncludeNames

---

---

# ImportSetFileName

|                   |                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetFileName specifies the name of a file to import.                                                                            |
| <b>Syntax</b>     | ImportSetFileName (Filename: <i>Character Expression</i> )                                                                           |
| <b>Parameters</b> |                                                                                                                                      |
| <i>Filename</i>   | Character Expression: Specifies the name of a file to import                                                                         |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select an option from the Data Type pop-up list, enter a filename. |
| <b>Example</b>    | To import a file named a:\data.str, the command is:<br><code>ImportSetFileName (Filename: "a:\data.str")</code>                      |
| <b>See Also</b>   | ♦ ImportSetFields                                                                                                                    |

---

# ImportSetFilePath

|                   |                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetFilePath specifies a Netware SQL file path.                                                        |
| <b>Syntax</b>     | ImportSetFilePath (FilePath: <i>Character Expression</i> )                                                  |
| <b>Parameters</b> |                                                                                                             |
| <i>FilePath</i>   | Character Expression: Specifies the path of a database file to import.                                      |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, Logon, enter file path.                   |
| <b>Example</b>    | To specify the file path W:\FILES, the command is:<br><code>ImportSetFilePath (FilePath: "w:\files")</code> |
| <b>See Also</b>   | ♦ ImportSetDictPath ♦ ImportSetPassword ♦ ImportSetUserID                                                   |

---

# ImportSetIncludeNames

|                   |                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetIncludeNames specifies whether to use field names as headings for imported database data.                                              |
| <b>Syntax</b>     | ImportSetIncludeNames (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                             |
| <b>Parameters</b> |                                                                                                                                                 |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to use field names as headings for imported database data.<br>0 Off!<br>1 On!          |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select an option from the Data Type pop-up list, Use field names as headings. |
| <b>Example</b>    | To use field names as headings for imported database data, the command is:<br><code>ImportSetIncludeNames (State: On!)</code>                   |
| <b>See Also</b>   | ♦ ImportSetFields                                                                                                                               |

---

# ImportSetPanel

|                   |                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetPanel specifies a DataPerfect panel to retrieve into the current document.                                                      |
| <b>Syntax</b>     | ImportSetPanel (PanelName: <i>Character Expression</i> )                                                                                 |
| <b>Parameters</b> |                                                                                                                                          |
| <i>PanelName</i>  | Character Expression: Specifies a DataPerfect panel name.                                                                                |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select DataPerfect from the Data Type pop-up list, enter a panel name. |
| <b>Example</b>    | To specify DataPerfect "panel1" to import, the command is:<br><code>ImportSetPanel (PanelName: "panel1")</code>                          |
| <b>See Also</b>   | ♦ ImportSetFileName ♦ ImportSetFields                                                                                                    |

---

# ImportSetPassword

|                   |                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetPassword specifies a password for importing NetWare SQL data.                                                                      |
| <b>Syntax</b>     | ImportSetPassword (Password: <i>Character Expression</i> )                                                                                  |
| <b>Parameters</b> |                                                                                                                                             |
| <i>Password</i>   | Character Expression: Specifies a password for importing NetWare SQL data.                                                                  |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, Select NetWare SQL from the Data Type pop-up list, Logon, enter password. |
| <b>Example</b>    | To use “bongo” as a password, the command is:<br><br>ImportSetPassword (Password: "bongo")                                                  |

---

# ImportSetQuery

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>          | ImportSetQuery specifies field conditions for a database import query.<br><br>IMPORTANT: For examples of valid conditions, choose Example from the Define Selection Conditions dialog box.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Syntax</b>           | ImportSetQuery (FieldName1: <i>Character Expression</i> ; FieldName2: <i>Character Expression</i> ; FieldName3: <i>Character Expression</i> ; Condition1Field1: <i>Character Expression</i> ; Condition1Field2: <i>Character Expression</i> ; Condition1Field3: <i>Character Expression</i> ; Condition2Field1: <i>Character Expression</i> ; Condition2Field2: <i>Character Expression</i> ; Condition2Field3: <i>Character Expression</i> ; Condition3Field1: <i>Character Expression</i> ; Condition3Field2: <i>Character Expression</i> ; Condition3Field3: <i>Character Expression</i> ; Condition4Field1: <i>Character Expression</i> ; Condition4Field2: <i>Character Expression</i> ; Condition4Field3: <i>Character Expression</i> ) |
| <b>Parameters</b>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>FieldName1</i>       | Character Expression (optional): Specifies the first field name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>FieldName2</i>       | Character Expression (optional): Specifies the second field name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>FieldName3</i>       | Character Expression (optional): Specifies the third field name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>Condition1Field1</i> | Character Expression (optional): Defines the first condition for field 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>Condition1Field2</i> | Character Expression (optional): Defines the first condition for field 2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                         |                                                                            |
|-------------------------|----------------------------------------------------------------------------|
| <i>Condition1Field3</i> | Character Expression (optional): Defines the first condition for field 3.  |
| <i>Condition2Field1</i> | Character Expression (optional): Defines the second condition for field 1. |
| <i>Condition2Field2</i> | Character Expression (optional): Defines the second condition for field 2. |
| <i>Condition2Field3</i> | Character Expression (optional): Defines the second condition for field 3. |
| <i>Condition3Field1</i> | Character Expression (optional): Defines the third condition for field 1.  |
| <i>Condition3Field2</i> | Character Expression (optional): Defines the third condition for field 2.  |
| <i>Condition3Field3</i> | Character Expression (optional): Defines the third condition for field 3.  |
| <i>Condition4Field1</i> | Character Expression (optional): Defines the fourth condition for field 1. |
| <i>Condition4Field2</i> | Character Expression (optional): Defines the fourth condition for field 2. |
| <i>Condition4Field3</i> | Character Expression (optional): Defines the fourth condition for field 3. |

---

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select an option from the Data Type pop-up list, enter field, Query, enter conditions.

---

**Example** To specify all the names up to and including “Ling” in the field Last Name, the command is:

`ImportSetQuery (FieldName1: "Last Name"; Condition1Field1: "-Ling")`

---

## ImportSetRange

---

**Purpose** ImportSetRange specifies a range of spreadsheet cells to import.

---

**Syntax** ImportSetRange (Range: *Character Expression*)

---

**Parameters**

*Range* Character Expression: Specifies a range of spreadsheet cells to import.

---

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select Spreadsheet from the Data Type pop-up list, enter a range of cells.

---

**Example** To import cells A1-F1, the command is:

`ImportSetRange (Range: "A1-F1")`



---

## ImportSetServer

|                   |                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetServer specifies a server for a database import.                                                                                                |
| <b>Syntax</b>     | ImportSetServer (ServerName: <i>Character Expression</i> )                                                                                               |
| <b>Parameters</b> |                                                                                                                                                          |
| <i>ServerName</i> | Character Expression: Specifies a server name.                                                                                                           |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select SQL Server from the Data Type pop-up list, Log On, enter a network server name. |
| <b>Example</b>    | To specify a server named Server 1, the command is:<br><br>ImportSetServer (ServerName: "Server 1")                                                      |

---

## ImportSetSource

|                   |                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ImportSetSource specifies the format of spreadsheet or database data to import.                                                                                                                                                                                          |
| <b>Syntax</b>     | ImportSetSource (SourceType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                       |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                          |
| <i>SourceType</i> | Numeric Equivalent or Enumerated Type: Specifies the format of data to import.<br><br>1 Spreadsheet!<br>33 dBase!<br>34 Paradox!<br>65 DataPerfect!<br>97 ASCII!<br>129 NetWare!<br>130 Oracle!<br>131 DB2!<br>132 SQLServer!<br>133 Sybase!<br>134 SQLBase!<br>135 XDB! |
| <b>Route</b>      | Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select an option from the Data Type pop-up list.                                                                                                                                                       |

---

**Example** To import spreadsheet data, the command is:

ImportSetSource (SourceType: Spreadsheet!)

---

## ImportSetSQLQuery

---

**Purpose** ImportSetSQLQuery specifies conditions for an SQL query.  
IMPORTANT: For examples of valid conditions, choose Example from the Define Selection Conditions dialog box.

---

**Syntax** ImportSetSQLQuery (SQLQuery: *Character Expression*)

---

### Parameters

*SQLQuery* Character Expression: Specifies conditions for an SQL query.

---

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select a SQL data type from the Data Type pop-up list, Log On, Query, enter conditions.

---

**Example** To select first and last names from the file SALES where the Units number is greater than 400, the command is:

ImportSetSQLQuery (SQLQuery: "SELECT first\_name, last\_name FROM Sales WHERE units > 400")

---

## ImportSetUserID

---

**Purpose** ImportSetUserID specifies an SQL database user identification.

---

**Syntax** ImportSetUserID (UserID: *Character Expression*)

---

### Parameters

*UserID* Character Expression: Specifies an SQL user identification.

---

**Route** Insert, Spreadsheet/Database, Import or Create Link or Edit Link, select NetWare SQL from the Data Type pop-up list, Log On, enter user identification.

---

**Example** To specify the user identification "Dana," the command is:

ImportSetUserID (UserID: "Dana")

---

**See Also**

♦ ImportSetDictPath ♦ ImportSetFilePath ♦ ImportSetPassword

---

## ImportUpdateDlg

---

**Purpose** ImportUpdateDlg displays the Update Import dialog box.

---

**Syntax** ImportUpdateDlg ()

---

**Route** Insert, Spreadsheet/Database, Update.

---

---

## Indent

---

**Purpose** Indent indents all lines in a paragraph without changing the current margin settings.

---

**Syntax** Indent ()

---

**Route** Press Indent (F7).

---

**See Also** ♦ ?Indent

---

---

## IndentLeftRight

---

**Purpose** IndentLeftRight indents equally both sides of a paragraph. Tab specifies the amount of the indent.

---

**Syntax** IndentLeftRight ()

---

**Route** Layout, Paragraph, Double Indent.

---

---

## IndexDefineDlg

---

**Purpose** IndexDefineDlg displays the Define Index dialog box.

---

**Syntax** IndexDefineDlg ()

---

---

## IndexDefinition

---

### Purpose

IndexDefinition specifies an index format.

---

### Syntax

IndexDefinition (Heading: *Numeric Equivalent* or *Enumerated Type*; Subheading: *Numeric Equivalent* or *Enumerated Type*; NumMode: *Numeric Equivalent* or *Enumerated Type*; PGCombine: *Numeric Equivalent* or *Enumerated Type*; Concordance: *Character Expression*; Page NumberFormat: *Character Expression*)

---

### Parameters

#### Heading

Numeric Equivalent or Enumerated Type: Specifies the style of an index heading.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TextBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 41 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutInStyle!

- 46 OutlineStyle!
- 47 LegalOutInStyle!
- 48 BulletsOutInStyle!
- 49 HeadingsOutInStyle!
- 50 Legal\_2OutInStyle!
- 51 NumbersOutInStyle!
- 52 Level1Style!
- 53 Level2Style!
- 54 Level3Style!
- 55 Level4Style!
- 56 Level5Style!
- 57 Level6Style!
- 58 Level7Style!
- 59 Level8Style!
- 60 Legal1Style!
- 61 Legal2Style!
- 62 Legal3Style!
- 63 Legal4Style!
- 64 Legal5Style!
- 65 Legal6Style!
- 66 Legal7Style!
- 67 Legal8Style!
- 68 Heading1Style!
- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

*Subheading*

Numeric Equivalent or Enumerated Type: Specifies an index subheading style. See the table under the Heading parameter for a list of numeric equivalents and enumerated types.

*NumMode*

Numeric Equivalent or Enumerated Type: Specifies the method of displaying a page number after a heading.

- 0 DefNone!
- 1 DefFollows!
- 2 DefParentheses!
- 3 DefFlushRight!
- 4 DefLeader!

*PGCombine*

Numeric Equivalent or Enumerated Type: Specifies whether sequential page numbers are combined (for example, 51-53) or remain separate (for example, 51, 52, 53).

- 0 NoCombine!
- 1 Combine!

*Concordance*  
*UserFormat*

Character Expression: Specifies the filename of a concordance file.

Character Expression: Specifies an index page number format. If you leave this parameter blank, page numbers are formatted as they are in your document.

---

**Route** Tools, Index, Define, select options.

---

**Example** To define an index, the command is:

```
IndexDefinition (Heading: Index1Style!; Subheading: Index2Style!; NumMode: DefFlushRight!;
PGCombine: Combine!; Concordance: ""; UserFormat: "")
```

---

**See Also** ♦ IndexMark ♦ ?IndexDefinition

---

## IndexMark

---

**Purpose** IndexMark marks text to index.

---

**Syntax** IndexMark (Heading: *Character Expression*; Subheading: *Character Expression*)

---

**Parameters**

*Heading* Character Expression: Specifies an index heading.

*Subheading* Character Expression: Specifies an index subheading.

---

**Route** Tools, Index, Mark, enter a heading and subheading, Mark.

---

**Example** To insert an index code for "Ham, baked," the command is:

```
IndexMark (Heading: "Ham"; Subheading: "baked")
```

---

**See Also** ♦ IndexDefinition

---

## InhibitInput

---

**Purpose** InhibitInput specifies whether to allow keyboard and menu input during macro execution.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

|                   |                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | InhibitInput (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                      |
| <b>Parameters</b> |                                                                                                                 |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: <ul style="list-style-type: none"> <li>0 Off!</li> <li>1 On!</li> </ul>  |
| <b>Example</b>    | To allow keyboard and menu input during the execution of a macro, the command is:<br>InhibitInput (State: Off!) |
| <b>See Also</b>   | ♦ Display                                                                                                       |

---

## InitialCodesStyleDlg

|                |                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | InitialCodesStyleDlg displays the Styles Editor dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | InitialCodesStyleDlg ()                                                                                                                                         |

---

## InsertDefaultBullet

|                |                                                                                 |
|----------------|---------------------------------------------------------------------------------|
| <b>Purpose</b> | InsertDefaultBullet inserts another bullet or number of the last type inserted. |
| <b>Syntax</b>  | InsertDefaultBullet ()                                                          |
| <b>Route</b>   | Press Ctrl+Shift+B.                                                             |

---

## InsertFilename

|                |                                                                     |
|----------------|---------------------------------------------------------------------|
| <b>Purpose</b> | InsertFilename inserts the current filename at the insertion point. |
| <b>Syntax</b>  | InsertFilename ()                                                   |

---

|                 |                          |
|-----------------|--------------------------|
| <b>Route</b>    | Insert, Other, Filename. |
| <b>See Also</b> | ♦ InsertFilenameWithPath |

---

---

## InsertFilenameWithPath

---

|                 |                                                                                      |
|-----------------|--------------------------------------------------------------------------------------|
| <b>Purpose</b>  | InsertFilenameWithPath inserts the current filename and path at the insertion point. |
| <b>Syntax</b>   | InsertFilenameWithPath ()                                                            |
| <b>Route</b>    | Insert, Other, Path and Filename.                                                    |
| <b>See Also</b> | ♦ InsertFilename                                                                     |

---

---

## InsertOtherCodesDlg

---

|                |                                                          |
|----------------|----------------------------------------------------------|
| <b>Purpose</b> | InsertOtherCodesDlg displays the Other Codes dialog box. |
| <b>Syntax</b>  | InsertOtherCodesDlg ()                                   |
| <b>Route</b>   | Layout, Line, Other Codes.                               |

---

---

## InvokeDialog

---

|                   |                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | InvokeDialog displays a specified dialog box but does not pause the macro to allow information to be entered.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | InvokeDialog (Dialog: <i>Macro Command</i> )                                                                                                                                                                          |
| <b>Parameters</b> |                                                                                                                                                                                                                       |
| <i>Dialog</i>     | Macro Command: Specifies a macro command which displays a dialog box.                                                                                                                                                 |

---



---

**Example** To invoke the Advance dialog box, the command is:  
InvokeDialog (Dialog: AdvanceDlg)

---

## Justification

---

**Purpose** Justification aligns text on right or left margins, on both margins, or centered between margins.

---

**Syntax** Justification (Justification: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Justification* Numeric Equivalent or Enumerated Type: Specifies text justification.

- 0 Left!
- 1 Full!
- 2 Center!
- 3 Right!
- 4 FullAll!
- 5 DecAlign!

---

**Route** Layout, Justification, select a type.

---

**Example** To center all subsequent lines of text, the command is:  
Justification (Justification: Center!)

---

**See Also** ♦ ?Justification

---

## JustifyAll

---

**Purpose** JustifyAll aligns all text to left and right margins.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** JustifyAll ()

---

## JustifyCenter

|                 |                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | JustifyCenter centers text between margins from the current paragraph forward.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | JustifyCenter ()                                                                                                                                                                   |
| <b>See Also</b> | ♦ JustifyFull ♦ JustifyLeft ♦ JustifyRight                                                                                                                                         |

---

## JustifyFull

|                 |                                                                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | JustifyFull aligns text along the left and right margins from the current paragraph forward.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | JustifyFull ()                                                                                                                                                                                   |
| <b>See Also</b> | ♦ JustifyCenter ♦ JustifyLeft ♦ JustifyRight                                                                                                                                                     |

---

## JustifyLeft

|                 |                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | JustifyLeft aligns text along the left margin from the current paragraph forward.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | JustifyLeft ()                                                                                                                                                                        |
| <b>See Also</b> | ♦ JustifyCenter ♦ JustifyFull ♦ JustifyRight                                                                                                                                          |

---

## JustifyRight

|                |                                                                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | JustifyRight aligns text along the right margin from the current paragraph forward.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                             |
|-----------------|---------------------------------------------|
| <b>Syntax</b>   | JustifyRight ()                             |
| <b>See Also</b> | ♦ JustifyCenter ♦ JustifyFull ♦ JustifyLeft |

---

## KeepTextTogetherDlg

---

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| <b>Purpose</b> | KeepTextTogetherDlg displays the Keep Text Together dialog box. |
| <b>Syntax</b>  | KeepTextTogetherDlg ()                                          |
| <b>Route</b>   | Layout, Page, Keep Text Together.                               |

---

## Kerning

---

|                   |                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Kerning selects or deselects Automatic Kerning in the Word Spacing and Letterspacing dialog box.            |
| <b>Syntax</b>     | Kerning (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                       |
| <b>Parameters</b> |                                                                                                             |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to turn on or turn off Kerning.<br>0 Off!<br>1 On! |
| <b>Route</b>      | Layout, Typesetting, Word/Letterspacing, select or deselect Automatic Kerning.                              |
| <b>Example</b>    | To turn on kerning, the command is:<br>Kerning (State: On!)                                                 |
| <b>See Also</b>   | ♦ ?Kerning                                                                                                  |

---

---

# KerningSpacing

---

**Purpose** KerningSpacing moves the character to the right of the insertion point a specified amount of space left or right.

---

**Syntax** KerningSpacing (KerningSpacingValue: *Measurement Expression*)

---

## Parameters

*KerningSpacingValue* Measurement Expression: Specifies the distance to move the character. A negative value moves a character left. A positive value moves a character right.

---

**Route** Layout, Typesetting, Manual Kerning.

---

**Example** To move a character .005i to the right, the command is:

KerningSpacing (KerningSpacingValue: .005i)

---

**See Also** ♦ Advance

---

# KeyboardSelect

---

**Purpose** KeyboardSelect selects a keyboard, such as WPWin 6.0 Keyboard, Equation Editor Keyboard, or WPDOS 6.0 Compatible keyboard.

---

**Syntax** KeyboardSelect (Keyboard: *Character Expression*)

---

## Parameters

*Keyboard* Character Expression: Specifies the name of a keyboard.

---

**Route** File, Preferences, Keyboard icon, select option.

---

**Example** To specify the CUA keyboard, the command is:

KeyboardSelect (Keyboard: "<WPWIN 6.0 Keyboard>")

---

**See Also** ♦ KeyboardSelectDlg

---

# KeyboardSelectDlg

---

|                 |                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | KeyboardSelectDlg displays the Keyboard Preferences dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | KeyboardSelectDlg ()                                                                                                                                                |
| <b>See Also</b> | ♦ KeyboardSelect                                                                                                                                                    |

---

# LabelsDefine

---

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | LabelsDefine creates a customized labels definition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Syntax</b>  | LabelsDefine (Filename: <i>Character Expression</i> ; LabelName: <i>Character Expression</i> ; LabelFlags: <i>Numeric Equivalent or Enumerated Type</i> ; SheetWidth: <i>Measurement Expression</i> ; SheetLength: <i>Measurement Expression</i> ; ColumnsPerPage: <i>Numeric Expression</i> ; RowsPerPage: <i>Numeric Expression</i> ; LabelWidth: <i>Measurement Expression</i> ; LabelLength: <i>Measurement Expression</i> ; LeftOffset: <i>Measurement Expression</i> ; TopOffset: <i>Measurement Expression</i> ; BetweenColumns: <i>Measurement Expression</i> ; BetweenRows: <i>Measurement Expression</i> ; MarginLeft: <i>Measurement Expression</i> ; MarginRight: <i>Measurement Expression</i> ; MarginTop: <i>Measurement Expression</i> ; MarginBottom: <i>Measurement Expression</i> ) |

---

## Parameters

|                       |                                                                                                                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Filename</i>       | Character Expression: Specifies the filename where customized labels are saved.                                                                                                                                                                                        |
| <i>LabelName</i>      | Character Expression: Specifies a label name.                                                                                                                                                                                                                          |
| <i>LabelFlags</i>     | Numeric Equivalent or Enumerated Type: Specifies whether the labels displayed in the Labels dialog box are for laser or tractor-fed printers, or both.<br><ol style="list-style-type: none"><li>1 Laser!</li><li>2 TractorFed!</li><li>3 LaserAndTractorFed!</li></ol> |
| <i>SheetWidth</i>     | Measurement Expression: Specifies the width of a page of labels.                                                                                                                                                                                                       |
| <i>SheetLength</i>    | Measurement Expression: Specifies the length of a page of labels.                                                                                                                                                                                                      |
| <i>ColumnsPerPage</i> | Numeric Expression: Specifies the number of columns on a page.                                                                                                                                                                                                         |
| <i>RowsPerPage</i>    | Numeric Expression: Specifies the number of rows on a page.                                                                                                                                                                                                            |

|                       |                                                                                                                           |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>LabelWidth</i>     | Measurement Expression: Specifies label width.                                                                            |
| <i>LabelLength</i>    | Measurement Expression: Specifies label length.                                                                           |
| <i>LeftOffset</i>     | Measurement Expression: Specifies the distance from the left edge of the page to the left edge of the first label column. |
| <i>TopOffset</i>      | Measurement Expression: Specifies the distance from the top edge of the page to the top edge of the first label row.      |
| <i>BetweenColumns</i> | Measurement Expression: Specifies the distance between label columns.                                                     |
| <i>BetweenRows</i>    | Measurement Expression: Specifies the distance between label rows.                                                        |
| <i>MarginLeft</i>     | Measurement Expression: Specifies the left margin of each label.                                                          |
| <i>MarginRight</i>    | Measurement Expression: Specifies the right margin of each label.                                                         |
| <i>MarginTop</i>      | Measurement Expression: Specifies the top margin of each label.                                                           |
| <i>MarginBottom</i>   | Measurement Expression: Specifies the bottom margin of each label.                                                        |

---

**Route**                      Layout, Page, Labels, Create, enter a label description, specify measurements, select label type.

---

**Example**                      To create a labels paper size/type named My Labels with 3 columns and 10 rows on an 8.5" x 11" page, the command is:

```
LabelsDefine (Filename: "WP_WP_US.LAB"; LabelName: "My Labels"; LabelFlags: Laser!; SheetWidth: 8.5"; SheetLength: 11"; ColumnsPerPage: 3; RowsPerPage: 10; LabelWidth: 2.63"; LabelLength: 1"; LeftOffset: 0.188"; TopOffset: 0.5"; BetweenColumns: 0.125"; BetweenRows: 0"; MarginLeft: 0"; MarginRight: 0"; MarginTop: 0"; MarginBottom: 0")
```

---

**See Also**                      ♦ LabelsFileDescription ♦ LabelsSelect

---

## LabelsDelete

---

**Purpose**                      LabelsDelete deletes a labels definition.

---

**Syntax**                      LabelsDelete (Filename: *Character Expression*; LabelName: *Character Expression*)

---

### Parameters

*Filename*                      Character Expression: Specifies the name of a labels file to delete from.

*LabelName*                      Character Expression: Specifies a label definition to delete.

---

|                |                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Layout, Page, Labels, select a labels definition in the Labels list box, Delete.                                                               |
| <b>Example</b> | To delete a labels definition named My Labels, the command is:<br><code>LabelsDelete (Filename: "WP_WP_US.LAB"; LabelName: "My Labels")</code> |

---

## LabelsDlg

---

|                |                                           |
|----------------|-------------------------------------------|
| <b>Purpose</b> | LabelsDlg displays the Labels dialog box. |
| <b>Syntax</b>  | LabelsDlg ()                              |
| <b>Route</b>   | Layout, Labels.                           |

---

## LabelsFileCreate

---

|                   |                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LabelsFileCreate creates a labels file.                                                                                                                |
| <b>Syntax</b>     | LabelsFileCreate (Filename: <i>Character Expression</i> )                                                                                              |
| <b>Parameters</b> |                                                                                                                                                        |
| <i>Filename</i>   | Character Expression: Specifies the path of a new labels file.                                                                                         |
| <b>Route</b>      | Layout, Labels, Change, Create, enter a filename.                                                                                                      |
| <b>Example</b>    | To create a labels file named CUSTOM.LAB in the C:\WPC20 directory, the command is:<br><code>LabelsFileCreate (Filename: "C:\WPC20\CUSTOM.LAB")</code> |
| <b>See Also</b>   | ♦ LabelsFileDescription ♦ LabelsFileSelect                                                                                                             |

---

---

## LabelsFileDescription

---

|                    |                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | LabelsFileDescription adds a description to an existing labels file.                                                                                                                 |
| <b>Syntax</b>      | LabelsFileDescription (Filename: <i>Character Expression</i> ; Description: <i>Character Expression</i> )                                                                            |
| <b>Parameters</b>  |                                                                                                                                                                                      |
| <i>Filename</i>    | Character Expression: Specifies the filename of a labels file.                                                                                                                       |
| <i>Description</i> | Character Expression: Specifies a labels file description.                                                                                                                           |
| <b>Route</b>       | Layout, Labels, Change, highlight a label file, Edit, enter a description.                                                                                                           |
| <b>Example</b>     | To add the description Label Library to a labels file named CUSTOM.LAB, the command is:<br><br>LabelsFileDescription (Filename: "C:\WPC20\CUSTOM.LAB"; Description: "Label Library") |
| <b>See Also</b>    | ♦ LabelsFileCreate ♦ LabelsFileSelect                                                                                                                                                |

---

## LabelsFileSelect

---

|                   |                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LabelsFileSelect selects a labels file.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | LabelsFileSelect (Filename: <i>Character Expression</i> )                                                                                       |
| <b>Parameters</b> |                                                                                                                                                 |
| <i>Filename</i>   | Character Expression: Specifies the filename of a labels file.                                                                                  |
| <b>Example</b>    | To select a labels file named CUSTOM.LAB, the command is:<br><br>LabelsFileSelect (Filename: "C:\WPC20\CUSTOM.LAB")                             |
| <b>See Also</b>   | ♦ LabelsFileDescription                                                                                                                         |



---

## LabelsOff

---

**Purpose** LabelsOff turns off labels mode.

---

**Syntax** LabelsOff ()

---

**Route** Layout, Labels, Off.

---

**See Also** ♦ LabelsFileSelect, LabelsSelect

---

## LabelsSelect

---

**Purpose** LabelsSelect selects a label definition.

---

**Syntax** LabelsSelect (Filename: *Character Expression*; LabelName: *Character Expression*)

---

### Parameters

*Filename* Character Expression: Specifies a label filename.

*LabelName* Character Expression: Specifies a label definition to select.

---

**Route** Layout, Labels, highlight a label definition in the Labels list box, Select.

---

**Example** To select a label definition named My Labels from the default WordPerfect labels file named WP\_WP\_US.LAB, the command is:

LabelsSelect (Filename: "C:\WPC20\WP\_WP\_US.LAB"; LabelName: "My Labels")

---

## Language

---

**Purpose** Language specifies the language used for a document or section of text.

---

**Syntax** Language (Language: *Character Expression*)

---

### Parameters

*Language* Character Expression: Two characters that specify the language of a document or section of text.

---

|                |                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Tools, Language, select language.                                                                     |
| <b>Example</b> | To specify Spanish as the language of a section of text, the command is:<br>Language (Language: "ES") |

---

## LanguageDlg

---

|                |                                               |
|----------------|-----------------------------------------------|
| <b>Purpose</b> | LanguageDlg displays the Language dialog box. |
| <b>Syntax</b>  | LanguageDlg ()                                |
| <b>Route</b>   | Tools, Language.                              |

---

## LastOpenedAdd

---

|                   |                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LastOpenedAdd places a filename at the top of the last opened filenames list on the File menu.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | LastOpenedAdd (Filename: <i>Character Expression</i> )                                                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                                        |
| <i>Filename</i>   | Character Expression: Specifies a filename.                                                                                                                                                            |
| <b>Example</b>    | To place TEST.DOC at the top of the last opened filenames list on the File menu, the command is:<br><br>LastOpenedAdd (Filename: "C:\WPWIN60\DOCSVTEST.DOC")                                           |

---

## Leading

---

|                |                                       |
|----------------|---------------------------------------|
| <b>Purpose</b> | Leading specifies leading adjustment. |
|----------------|---------------------------------------|

---

|                   |                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | Leading (Adjustment: <i>Measurement Expression</i> )                                                                 |
| <b>Parameters</b> |                                                                                                                      |
| <i>Adjustment</i> | Measurement Expression: Specifies a leading adjustment.                                                              |
| <b>Route</b>      | Layout, Typesetting, Word/Letterspacing, select Adjust Leading, specify a measurement in the Between Lines edit box. |
| <b>Example</b>    | To specify a leading adjustment of .1 inches, the command is:<br>Leading (Adjustment: .1")                           |

---

## LineCreateDlg

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <b>Purpose</b> | LineCreateDlg displays the Create Graphics Line dialog box. |
| <b>Syntax</b>  | LineCreateDlg ()                                            |
| <b>Route</b>   | Graphics, Custom Line.                                      |

---

## LineEditDlg

|                 |                                                                                                                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | LineEditDlg displays the Edit Graphics Line dialog box.<br>IMPORTANT: If there is no graphics line to edit, this command terminates the macro or redirects macro execution to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | LineEditDlg ()                                                                                                                                                                                                |
| <b>Route</b>    | Graphics, Edit Line.                                                                                                                                                                                          |
| <b>See Also</b> | ♦ LineCreateDlg                                                                                                                                                                                               |

---

## LineHeight

|                |                                   |
|----------------|-----------------------------------|
| <b>Purpose</b> | LineHeight specifies line height. |
|----------------|-----------------------------------|

**IMPORTANT:** Line height is measured from baseline to baseline in single-spaced text. Auto line height configures line height as the point size of the current font plus two points.

---

**Syntax** LineHeight (Height: *Measurement Expression* or *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Height* Measurement Expression or Numeric Equivalent or Enumerated Type: Specifies a fixed line height. To specify Auto Line Height, use the numeric equivalent or enumerated type.

0 Auto!

---

**Route** Layout, Line, Height, Auto or Fixed, enter a measurement.

---

**Example** To specify a line height of .275 inches, the command is:

LineHeight (Height: .275")

---

## LineHeightDlg

---

**Purpose** LineHeightDlg displays the Line Height dialog box.

---

**Syntax** LineHeightDlg ()

---

**Route** Layout, Line, Height.

---

**See Also** ♦ LineHeight

---

---

## LineHyphenationDlg

---

**Purpose** LineHyphenationDlg displays the Line Hyphenation dialog box.

---

**Syntax** LineHyphenationDlg ()

---

**Route** Layout, Line, Hyphenation.

---

# LineNumbering

---

|                |                                        |
|----------------|----------------------------------------|
| <b>Purpose</b> | LineNumbering turns on Line Numbering. |
|----------------|----------------------------------------|

---

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | LineNumbering (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; RestartAtNewPage: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; CountBlankLines: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; NumberColumns: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; PositionRelativeToLeftEdge: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Position: <i>Measurement Expression</i> ; FirstPrintingNumber: <i>Numeric Expression</i> ; NumberInterval: <i>Numeric Expression</i> ; Default: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; FontFace: <i>Character Expression</i> ; FontSize: <i>Numeric Expression</i> ; Red: <i>Numeric Expression</i> ; Green: <i>Numeric Expression</i> ; Blue: <i>Numeric Expression</i> ; Shade: <i>Numeric Expression</i> ; AttributesOn: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                   |  |
|-------------------|--|
| <b>Parameters</b> |  |
|-------------------|--|

|                                   |                                                                                                                                                                                                                                   |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>State</i>                      | Numeric Equivalent or Enumerated Type: Specifies whether Line Numbering is on or off.<br><br>0 Off!<br>1 On!                                                                                                                      |
| <i>RestartAtNewPage</i>           | Numeric Equivalent or Enumerated Type: Specifies whether Line Numbering is continuous or restarted on each page.<br><br>0 DontRestartNumbersEachPage!<br>1 RestartNumbersEachPage!                                                |
| <i>CountBlankLines</i>            | Numeric Equivalent or Enumerated Type: Specifies whether blank lines are counted in Line Numbering.<br><br>0 DontCountBlankLines!<br>1 CountBlankLines!                                                                           |
| <i>NumberColumns</i>              | Numeric Equivalent or Enumerated Type: Specifies whether lines in newspaper columns are numbered.<br><br>0 DontNumberAllNewspaperColumns!<br>1 NumberAllNewspaperColumns!                                                         |
| <i>PositionRelativeToLeftEdge</i> | Numeric Equivalent or Enumerated Type: Specifies whether the measurement given in the Position parameter is measured from the left edge of a page or margin.<br><br>0 PosNumberFromLeftMargin!<br>1 PosNumberFromLeftEdgeOfPaper! |
| <i>Position</i>                   | Measurement Expression: Specifies where the line number prints.                                                                                                                                                                   |

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>NumberInterval</i> | Numeric Expression: Specifies the numbering interval.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>Default</i>        | Numeric Equivalent or Enumerated Type: Specifies whether a line numbering font uses a document's initial font attributes.<br>0 UseDefaultValues!<br>1 DontUseDefaultValues!                                                                                                                                                                                                                                                                                                                                  |
| <i>FontFace</i>       | Character Expression: Specifies the line number font. Used only when the Default parameter is set to DontUseDefaultValues! or 1.                                                                                                                                                                                                                                                                                                                                                                             |
| <i>FontSize</i>       | Numeric Expression: Specifies the point size of a line number font. Used only when the Default parameter is set to DontUseDefaultValues! or 1.                                                                                                                                                                                                                                                                                                                                                               |
| <i>Red</i>            | Numeric Expression: Represents the amount of red in a color. Values range from 0 to 255. Used only when the Default parameter is set to DontUseDefaultValues! or 1. The values of a selected color are available in the Define Color Printing Palette dialog box.                                                                                                                                                                                                                                            |
| <i>Green</i>          | Numeric Expression: Represents the amount of green in a color. Values range from 0 to 255. Used only when the Default parameter is set to DontUseDefaultValues! or 1. The values of a selected color are available in the Define Color Printing Palette dialog box.                                                                                                                                                                                                                                          |
| <i>Blue</i>           | Numeric Expression: Represents the amount of blue in a color. Values range from 0 to 255. Used only when the Default parameter is set to DontUseDefaultValues! or 1. The values of a selected color are available in the Define Color Printing Palette dialog box.                                                                                                                                                                                                                                           |
| <i>Shade</i>          | Numeric Expression: Represents the shading percentage of a color. Values range from 0 to 100. Used only when the Default parameter is set to DontUseDefaultValues! or 1.                                                                                                                                                                                                                                                                                                                                     |
| <i>AttributesOn</i>   | Numeric Equivalent or Enumerated Type: Specifies attributes and relative size of a font. Attributes are separated by semicolons and can be enclosed in a pair of braces ({}). Used only when the Default parameter is set to DontUseDefaultValues! or 1.<br>0 ExtraLarge!<br>1 VeryLarge!<br>2 Large!<br>3 Small!<br>4 Fine!<br>5 Superscript!<br>6 Subscript!<br>7 Outline!<br>8 Italics!<br>9 Shadow!<br>10 Redline!<br>11 DoubleUnderline!<br>12 Bold!<br>13 Strikeout!<br>14 Underline!<br>15 SmallCaps! |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Line, Numbering, select options in the Line Numbering dialog box.                                                                                                                                                                                                                                                                                                                                                    |
| <b>Example</b>  | To number lines, the command is:<br><br><pre>LineNumbering (State: On!; RestartAtNewPage: RestartNumbersEachPage!; CountBlankLines: DontCountBlankLines!; NumberColumns: DontNumberAllNewspaperColumns!; PositionRelativeToLeftEdge: PosNumberFromLeftMargin!; Position: 1"; FirstPrintingNumber: 1; NumberInterval: 5; Default: DontUseDefaultValues!; AttributesOn: {ExtraLarge!; Shadow!; DoubleUnderline!; Bold!})</pre> |
| <b>See Also</b> | ♦ LineNumberSet ♦ LineNumberingMethod ♦ ?LineNumbering ♦ ?LineNumCountBlankLines ♦ ?StartingLineNumber                                                                                                                                                                                                                                                                                                                       |

---

## LineNumberingDlg

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| <b>Purpose</b>  | LineNumberingDlg displays the Line Numbering dialog box. |
| <b>Syntax</b>   | LineNumberingDlg ()                                      |
| <b>Route</b>    | Layout, Line, Numbering.                                 |
| <b>See Also</b> | ♦ LineNumbering                                          |

---

## LineNumberingMethod

|                |                                                                                             |
|----------------|---------------------------------------------------------------------------------------------|
| <b>Purpose</b> | LineNumberingMethod specifies a method to number lines.                                     |
| <b>Syntax</b>  | LineNumberingMethod (NumberingMethod: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

### Parameters

|                        |                                                                            |
|------------------------|----------------------------------------------------------------------------|
| <i>NumberingMethod</i> | Numeric Equivalent or Enumerated Type: Specifies a method to number lines. |
| 0                      | Numbers!                                                                   |
| 1                      | LowerLetters!                                                              |
| 2                      | UpperLetters!                                                              |
| 3                      | LowerRoman!                                                                |
| 4                      | UpperRoman!                                                                |

|                 |                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Line, Numbering, Numbering Method, select an option.                                                                       |
| <b>Example</b>  | To specify uppercase letters as the Line Numbering method, the command is:<br>LineNumberingMethod (NumberingMethod: UpperLetters!) |
| <b>See Also</b> | ♦ LineNumberSet ♦ LineNumbering ♦ ?LineNumberMethod                                                                                |

---

## LineNumberSet

|                   |                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineNumberSet specifies the first line number on a page.                                    |
| <b>Syntax</b>     | LineNumberSet (Number: <i>Numeric Expression</i> )                                          |
| <b>Parameters</b> |                                                                                             |
| <i>Number</i>     | Numeric Expression: Specifies a first line number.                                          |
| <b>Route</b>      | Layout, Line, Numbering, Starting Line Number, enter a number.                              |
| <b>Example</b>    | To specify the first line on a page as line 5, the command is:<br>LineNumberSet (Number: 5) |
| <b>See Also</b>   | ♦ LineNumbering ♦ LineNumberingMethod                                                       |

---

## LineSegmentColor

|                   |                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentColor specifies the color of a line segment.<br>IMPORTANT: This command must be used in conjunction with other commands, such as LineSegmentCreate and LineSegmentEnd.                   |
| <b>Syntax</b>     | LineSegmentColor (ColorName: <i>Character Expression</i> ; Red: <i>Numeric Expression</i> ; Green: <i>Numeric Expression</i> ; Blue: <i>Numeric Expression</i> ; Shade: <i>Numeric Expression</i> ) |
| <b>Parameters</b> |                                                                                                                                                                                                     |
| <i>ColorName</i>  | Character Expression: Specifies a color name.                                                                                                                                                       |



|              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| <i>Red</i>   | Numeric Expression: Specifies the amount of red in a color. Values range from 0 to 255.      |
| <i>Green</i> | Numeric Expression: Specifies the amount of green in a color. Values range from 0 to 255.    |
| <i>Blue</i>  | Numeric Expression: Specifies the amount of blue in a color. Values range from 0 to 255.     |
| <i>Shade</i> | Numeric Expression: Specifies the shading percentage of a color. Values range from 0 to 100. |

---

**Route** Graphics, Custom Line, Line Styles, select Line Style, Edit, Color, select a color.

---

**Example** To specify a blue line segment, the command is:

```
LineSegmentColor (ColorName: "Blue"; Red: 0; Green: 0; Blue: 255; Shade: 100)
```

---

**See Also**

- ♦ LineSegmentCreate ♦ LineStyleEdit ♦ LineSegmentEnd ♦ LineStyleEnd
- ♦ ?LineSegmentBlue ♦ ?LineSegmentGreen ♦ ?LineSegmentRed
- ♦ ?LineSegmentShading

---

## LineSegmentCreate

---

**Purpose** LineSegmentCreate creates a line segment with multiple lines, or a stacked line style. **IMPORTANT:** This command must be used in conjunction with other commands, such as LineStyleEdit.

---

**Syntax** LineSegmentCreate(Position: *Numeric Expression*)

---

### Parameters

*Position* Numeric Expression: Specifies where a line segment is created or stacked. The first (inside) position is 0.

---

**Route** Graphics, Custom Line, Line Styles, Create, Name, Add.

---

**Example** To stack a line in the first (inside) position, the command is:

```
LineSegmentCreate (Position: 0)
```

---

**See Also**

- ♦ LineSegmentDelete ♦ LineSegmentEnd ♦ LineStyleEdit ♦ TableEdit

---

---

## LineSegmentDelete

---

|                   |                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentDelete deletes a stacked line segment.<br><br>IMPORTANT: This command must be used in conjunction with other commands, such as LineStyleEdit and LineSegmentEnd. |
| <b>Syntax</b>     | LineSegmentDelete (Position: <i>Numeric Expression</i> )                                                                                                                    |
| <b>Parameters</b> |                                                                                                                                                                             |
| <i>Position</i>   | Numeric Expression: Specifies where to delete a stacked line. The first (inside) position is 0.                                                                             |
| <b>Route</b>      | Graphics, Custom Line, Line Styles, select a line style, Edit, Del.                                                                                                         |
| <b>Example</b>    | To delete an inside stacked line of a border, the command is:<br><br>LineSegmentDelete (Position: 0)                                                                        |
| <b>See Also</b>   | ♦ LineSegmentCreate ♦ LineStyleEdit ♦ LineStyleEnd                                                                                                                          |

---

## LineSegmentEdit

---

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentEdit edits a line segment.<br><br>IMPORTANT: This command must be followed by LineSegmentEnd.              |
| <b>Syntax</b>     | LineSegmentEdit (Position: <i>Numeric Expression</i> )                                                                |
| <b>Parameters</b> |                                                                                                                       |
| <i>Position</i>   | Numeric Expression: Specifies a line to edit. The first (inside) position is 0.                                       |
| <b>Route</b>      | Graphics, Custom Line, Line Style, select a line style, Edit.                                                         |
| <b>Example</b>    | To edit an inside stacked line, the command is:<br><br>LineSegmentEdit (Position: 0)<br>LineSegmentEnd (State: Save!) |
| <b>See Also</b>   | ♦ LineSegmentColor ♦ LineSegmentEnd ♦ LineSegmentSpacing<br>♦ LineSegmentWidth                                        |

---

# LineSegmentEnd

|                   |                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentEnd ends a series of commands modifying a line segment.<br>IMPORTANT: This command must be preceded by LineSegmentCreate or LineSegmentEdit. |
| <b>Syntax</b>     | LineSegmentEnd (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                            |
| <b>Parameters</b> |                                                                                                                                                         |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to save changes to a line segment.<br>0 Cancel!<br>1 Save!                                     |
| <b>Route</b>      | OK or Cancel in the Edit Line Style (or Create Line Style) dialog box.                                                                                  |
| <b>Example</b>    | To create and save a line segment, the command is:<br><br>LineSegmentCreate (Position: 0)<br>LineSegmentEnd (State: Save!)                              |
| <b>See Also</b>   | ♦ LineSegmentCreate ♦ LineSegmentEdit                                                                                                                   |

---

# LineSegmentSetUser

|                   |                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentSetUser specifies a number of dash/space lengths in a custom line segment.<br><br>IMPORTANT: This command must be followed by the number of LineSegmentSetUserData commands for the number of dash/space lengths specified, and by LineSegmentSetUserEnd. |
| <b>Syntax</b>     | LineSegmentSetUser (OnOffs: <i>Numeric Expression</i> )                                                                                                                                                                                                              |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                      |
| <i>OnOffs</i>     | Numeric Expression: Specifies a number of dash/space lengths in a custom line segment.                                                                                                                                                                               |
| <b>Route</b>      | Graphics, Custom Line, Line Styles, highlight a style, Edit, Custom, set spacing.                                                                                                                                                                                    |

---

**Example** To create a line segment with one dash length and one space length, the command sequence is:

```
LineSegmentSetUser (OnOffs: 2)
LineSegmentSetUserData (SegSize: 0.2")
LineSegmentSetUserData (SegSize: 0.1")
LineSegmentSetUserEnd ()
```

---

**See Also** ♦ LineSegmentSetUserData ♦ LineSegmentSetUserEnd ♦ LineSegmentSetWPG2

---

## LineSegmentSetUserData

---

**Purpose** LineSegmentSetUserData specifies a dash or space length in a custom line pattern. **IMPORTANT:** This command must be preceded by LineSegmentUser and followed by LineSegmentEnd.

---

**Syntax** LineSegmentSetUserData (SegSize: *Measurement Expression*)

---

### Parameters

*SegSize* Measurement Expression: Specifies a dash or space length in a custom line pattern.

---

**Route** Graphics, Custom Line, Line Styles, highlight a style, Edit, Custom, Set Spacing.

---

**Example** To create a line segment with one dash length and one space length, the command sequence is:

```
LineSegmentSetUser (OnOffs: 2)
LineSegmentSetUserData (SegSize: 0.2")
LineSegmentSetUserData (SetSize: 0.1")
LineSegmentSetUserEnd ()
```

---

**See Also** ♦ LineSegmentSetUser ♦ LineSegmentSetUserEnd ♦ LineSegmentSetWPG2

---

## LineSegmentSetUserEnd

---

**Purpose** LineSegmentSetUserEnd marks the end of a custom line pattern command sequence.

---

**Syntax** LineSegmentSetUserEnd ()

---

**Route** Graphics, Custom Line, Line Styles, highlight a line style, Edit, Custom.

---

**See Also** ♦ LineSegmentSetUser ♦ LineSegmentSetUserData ♦ LineSegmentSetWPG2

---

---

## LineSegmentSetWPG2

---

**Purpose** LineSegmentSetWPG2 sets a line segment pattern to a predefined WPG2 line pattern.

---

**Syntax** LineSegmentSetWPG2 (Style: *Numeric Expression*; LineStyle: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Style* Numeric Expression: Specifies a pattern set. The only valid entry is 0.

*LineStyle* Numeric Equivalent or Enumerated Type: Specifies which line pattern to use for the selected set.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!
- 9 ButtonBottomRightLine!
- 126 TableDefaultLine!
- 127 NoLine!

---

**Route** Graphics, Custom Line, Line Styles, highlight a line style, Edit (or Create, enter a style name), Pattern, select a predefined pattern.

---

**Example** To specify the Single Line pattern for a line segment, the command is:

```
LineSegmentSetWPG2 (Style: 0; LineStyle: SingleLine!)
```

---

**See Also** ♦ LineSegmentSetUser ♦ LineSegmentSetUserData ♦ LineSegmentSetUserEnd  
♦ ?LineSegmentWPG2Pattern

---

---

# LineSegmentSpacing

---

|                   |                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentSpacing sets the distance between line segments.                                                                                              |
| <b>Syntax</b>     | LineSegmentSpacing (Spacing: <i>Measurement Expression</i> )                                                                                             |
| <b>Parameters</b> |                                                                                                                                                          |
| <i>Spacing</i>    | Measurement Expression: Specifies a distance between line segments.                                                                                      |
| <b>Route</b>      | Graphics, Custom Line, Line Styles, select a line style, Edit (or Create, enter a style name), specify a measurement in the Spacing Below Line text box. |
| <b>Example</b>    | To set the distance between line segments at 0.5 inches, the command is:<br><code>LineSegmentSpacing (Spacing: 0.5")</code>                              |
| <b>See Also</b>   | ♦ ?LineSegmentSpacing                                                                                                                                    |

---

# LineSegmentWidth

---

|                   |                                                                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSegmentWidth specifies the width or thickness of a graphics line segment.<br>IMPORTANT: This command must be preceded by LineSegmentCreate or LineSegmentEdit and followed by LineSegmentEnd.            |
| <b>Syntax</b>     | LineSegmentWidth (Width: <i>Measurement Expression</i> )                                                                                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                                              |
| <i>Width</i>      | Measurement Expression: Specifies the width or thickness of a graphics line segment.                                                                                                                         |
| <b>Route</b>      | Graphics, Custom Line, Line Styles, highlight a line style, Edit (or Create, enter a style name), specify a measurement in the Width text box.                                                               |
| <b>Example</b>    | To set the width of a line segment at 0.2 inches, the command is:<br><code>LineSegmentEdit (Position: 0)</code><br><code>LineSegmentWidth (Width: 0.2")</code><br><code>LineSegmentEnd (State: Save!)</code> |
| <b>See Also</b>   | ♦ LineSegmentCreate ♦ LineSegmentEdit ♦ LineSegmentEnd<br>♦ ?LineSegmentWidth                                                                                                                                |

---

## LineSpacing

---

|                   |                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineSpacing specifies the vertical spacing between lines.<br>IMPORTANT: When this command is recorded, the tenths position is used; when entering this command in a macro, tenths is optional. |
| <b>Syntax</b>     | LineSpacing (Spacing: <i>Numeric Expression</i> )                                                                                                                                              |
| <b>Parameters</b> |                                                                                                                                                                                                |
| <i>Spacing</i>    | Numeric Expression: Specifies line spacing.                                                                                                                                                    |
| <b>Route</b>      | Layout, Line, Spacing, specify a line spacing.                                                                                                                                                 |
| <b>Example</b>    | To specify single-spaced line spacing, the command is:<br><code>LineSpacing (Spacing: 1.0)</code>                                                                                              |
| <b>See Also</b>   | ♦ ?LineSpacing                                                                                                                                                                                 |

---

## LineSpacingDlg

---

|                 |                                                      |
|-----------------|------------------------------------------------------|
| <b>Purpose</b>  | LineSpacingDlg displays the Line Spacing dialog box. |
| <b>Syntax</b>   | LineSpacingDlg ()                                    |
| <b>Route</b>    | Layout, Line, Spacing.                               |
| <b>See Also</b> | ♦ LineSpacing                                        |

---

## LineStyleCopy

---

|                |                                                                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | LineStyleCopy copies attributes from one line to another.                                                                                                                                                                                               |
| <b>Syntax</b>  | LineStyleCopy (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , FromLibrary: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , ToLibrary: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , DestName: <i>Character Expression</i> ) |

---

## Parameters

|                    |                                                                                                                                                                                                                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i>       | Numeric Equivalent or Enumerated Type: Specifies a line to copy.<br>0 SingleLine!<br>1 DoubleLine!<br>2 DashedLine!<br>3 DottedLine!<br>4 ThickLine!<br>5 ExtraThickLine!<br>6 ThinThickLine!<br>7 ThickThinLine!<br>8 ButtonTopLeftLine!<br>9 ButtonBottomRightLine!<br>126 TableDefaultLine!<br>127 NoLine! |
| <i>FromLibrary</i> | Numeric Equivalent or Enumerated Type: Specifies a library from which to copy a line.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                                                                                                                                              |
| <i>ToLibrary</i>   | Numeric Equivalent or Enumerated Type: Specifies a library to which a line is copied.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                                                                                                                                              |
| <i>DestName</i>    | Character Expression: Specifies a new line name.                                                                                                                                                                                                                                                              |

---

## Route

Graphics, Custom Line, Line Styles, select a style, Options, Copy, enter a new style name.

---

## Example

To copy the attributes of a line named Single Line in a personal library to a newly created line named New Line in the current document, the command is:

```
LineStyleCopy (Style: SingleLine!; FromLibrary: PersonalLibrary!; ToLibrary: CurrentDoc!; DestName: "New Line")
```

---

## See Also

♦ GraphicsLineCreate ♦ GraphicsLineEdit ♦ GraphicsLineEnd

---

# LineStyleCreate

---

## Purpose

LineStyleCreate creates a line style.

IMPORTANT: This command must be used in conjunction with other commands, such as LineStyleEnd.



|                   |                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | LineStyleCreate (Name: <i>Character Expression</i> ; Library: <i>Numeric Equivalent or Enumerated Type</i> )                                                 |
| <b>Parameters</b> |                                                                                                                                                              |
| <i>Name</i>       | Character Expression: Specifies a new line style name.                                                                                                       |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: <ul style="list-style-type: none"> <li>0 CurrentDoc!</li> <li>1 PersonalLibrary!</li> <li>2 SharedLibrary!</li> </ul> |
| <b>Route</b>      | Graphics, Custom Lines, Line Styles, Create, enter a style name. Line Style is also available in Tables and Graphics Boxes.                                  |
| <b>Example</b>    | To create a line style named Custom, the command is: <pre>LineStyleCreate (Name: "Custom"; Library: CurrentDoc!)</pre>                                       |
| <b>See Also</b>   | ♦ LineStyleEnd                                                                                                                                               |

---

## LineStyleDelete

|                   |                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineStyleDelete deletes a user-defined line style.<br>IMPORTANT: WordPerfect line styles cannot be deleted. |
| <b>Syntax</b>     | LineStyleDelete (Name: <i>Character Expression</i> )                                                        |
| <b>Parameters</b> |                                                                                                             |
| <i>Name</i>       | Character Expression: Specifies a line style to delete.                                                     |
| <b>Route</b>      | Graphics, Graphics Styles, Line, select a line style, Options, Delete.                                      |
| <b>Example</b>    | To delete a line style named Custom, the command is: <pre>LineStyleDelete (Name: "Custom")</pre>            |

---

# LineStyleEdit

---

## Purpose

LineStyleEdit edits a line style.

**IMPORTANT:** This command is used in conjunction with other commands, such as LineStyleEnd.

---

## Syntax

LineStyleEdit (Style: *Numeric Equivalent* or *Enumerated Type*, Library: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

### *Style*

Numeric Equivalent or Enumerated Type: Specifies a line style to edit.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!
- 9 ButtonBottomRightLine!
- 126 TableDefaultLine!
- 127 NoLine!

### *Library*

Numeric Equivalent or Enumerated Type: Specifies the library where a line style is stored.

- 0 CurrentDoc!
  - 1 PersonalLibrary!
  - 2 SharedLibrary!
- 

## Route

Graphics, Graphics Styles, Line, select a line style, Edit.

---

## Example

To edit the Thick Line line style, the command is:

LineStyleEdit (Style: ThickLine!; Library: CurrentDoc!)

---

## See Also

- ♦ LineStyleEnd
-

---

# LineStyleEnd

|                   |                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineStyleEnd ends a series of commands that change a line style.<br><br>IMPORTANT: This command must be preceded by commands which change a line style, such as LineStyleCreate or LineStyleEdit. |
| <b>Syntax</b>     | LineStyleEnd (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                        |
| <b>Parameters</b> |                                                                                                                                                                                                   |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to save line style changes.<br><br>0 Cancel!<br>1 Save!                                                                                  |
| <b>Route</b>      | Graphics, Graphics Styles, Line, Edit or Create, select options.                                                                                                                                  |
| <b>Example</b>    | To save line style changes, the command is:<br><br>LineStyleEnd (State: Save!)                                                                                                                    |
| <b>See Also</b>   | ♦ LineStyleCreate ♦ LineStyleEdit                                                                                                                                                                 |

---

# LineStyleName

|                   |                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | LineStyleName renames a line style.<br><br>IMPORTANT: Line styles shipped with the program cannot be renamed. This command must be preceded by LineStyleCreate or LineStyleEdit and followed by LineStyleEnd. |
| <b>Syntax</b>     | LineStyleName (Style: <i>Character Expression</i> )                                                                                                                                                           |
| <b>Parameters</b> |                                                                                                                                                                                                               |
| <i>Style</i>      | Character Expression: Specifies a line style.                                                                                                                                                                 |
| <b>Route</b>      | Graphics, Graphics Styles, Line, select a line style, Options, Rename, enter a new style name.                                                                                                                |
| <b>Example</b>    | To rename a line style named Custom, the command is:<br><br>LineStyleName (Style: "Custom")                                                                                                                   |

---

**See Also**

◆ LineStyleCreate ◆ LineStyleEdit ◆ LineStyleEnd

---

## LineStyleRetrieve

---

**Purpose**

LineStyleRetrieve retrieves a line style into a library.

---

**Syntax**

LineStyleRetrieve (Filename: *Character Expression*; Style: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*Filename*

Character Expression: Specifies a filename containing a line style to retrieve.

*Style*

Numeric Equivalent or Enumerated Type:

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route**

This feature is available in Tables, Graphics Boxes, and Graphics Lines. Choose Retrieve from the Options drop-down list in the Line Styles dialog box.

---

**Example**

To retrieve a line style named CustomIn into a personal library, the command is:

LineStyleRetrieve (Filename: "CustomIn"; Style: PersonalLibrary!)

---

## LineStyleSave

---

**Purpose**

LineStyleSave saves a line style library.

---

**Syntax**

LineStyleSave (Filename: *Character Expression*; Style: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*Filename*

Character Expression: Specifies the name of a file in which to save the line style library.

*Style*

Numeric Equivalent or Enumerated Type: Specifies a line style library.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

|                |                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | This feature is available in Table, Graphics Boxes, and Graphics Lines. Choose Save As from the Options drop-down list in the Line Styles dialog box.   |
| <b>Example</b> | To save a current document line style library with the filename CURRENT, the command is:<br><br>LineStyleSave (Filename: "CURRENT"; Style: CurrentDoc!) |

---

## LinkUpdate

|                |                                     |
|----------------|-------------------------------------|
| <b>Purpose</b> | LinkUpdate updates a linked object. |
| <b>Syntax</b>  | LinkUpdate ()                       |
| <b>Route</b>   | Edit, Links, Update Now.            |

---

## ListDefineDlg

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | ListDefineDlg displays the Define List dialog box. |
| <b>Syntax</b>  | ListDefineDlg ()                                   |
| <b>Route</b>   | Tools, List, Define button.                        |

---

## ListDefinition

|                |                                                                                                                                                                                                                                                                                                                             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ListDefinition selects a list definition and defines the page where the list is generated. The item must exist in the defined lists or be created by ListDefinitionCreate before it can be selected.                                                                                                                        |
| <b>Syntax</b>  | ListDefinition (ListName: <i>Character Expression</i> ; Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Numbering: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Counter: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; NumberingFormat: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

## Parameters

*ListName*

Character Expression: Names or renames a selected list definition.

*Style*

Numeric Equivalent or Enumerated Type: Specifies a list style.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TextBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutInStyle!
- 46 OutlineStyle!
- 47 LegalOutInStyle!
- 48 BulletsOutInStyle!
- 49 HeadingsOutInStyle!
- 50 Legal\_2OutInStyle!
- 51 NumbersOutInStyle!
- 52 Level1Style!
- 53 Level2Style!
- 54 Level3Style!
- 55 Level4Style!
- 56 Level5Style!
- 57 Level6Style!
- 58 Level7Style!

- 59 Level8Style!
- 60 Legal1Style!
- 61 Legal2Style!
- 62 Legal3Style!
- 63 Legal4Style!
- 64 Legal5Style!
- 65 Legal6Style!
- 66 Legal7Style!
- 67 Legal8Style!
- 68 Heading1Style!
- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

*Numbering*

Numeric Equivalent or Enumerated Type: Specifies a numbering mode when a list is generated.

- 0 DefNone!
- 1 DefFollows!
- 2 DefParentheses!
- 3 DefFlushRight!
- 4 DefLeader!

*Counter*

Numeric Equivalent or Enumerated Type: Specifies a counter type.

- 0 FigureBox!
- 1 TableBox!
- 2 TextBox!
- 3 UserBox!
- 4 EquationBox!
- 5 ButtonBox!
- 6 WatermarkImageBox!
- 7 InlineEquationBox!
- 127 NoBox!

*NumberingFormat*

Character Expression: Specifies characters used to number a list.

**Route**

Tools, List, Define, select a list definition, Insert.

**Example**

To select a list definition named Subject List, the command is:

ListDefinition (ListName: "Subject List")

**See Also**

♦ ListDefinitionCreate ♦ ?ListDefinition

---

## ListDefinitionCreate

---

**Purpose**

ListDefinitionCreate creates a list definition.

IMPORTANT: This command is followed by other commands such as ListDefinitionMode and ListDefinitionStyle.

**Syntax**

ListDefinitionCreate (ListName: *Character Expression*)

**Parameters***ListName*

Character Expression: Names a new list definition.

**Route**

Tools, List, Define, Create, select options in the Create List dialog box.

**Example**

To create a list definition named Subject List which specifies a List style and numbering with dot leaders following entries, includes figure boxes, and specifies the document page numbering method, the command is:

```
MarkList ()
ListDefinitionCreate (ListName: "Subject List")
ListDefinitionMode (ListName: "Subject List"; Numbering: DefLeader!)
ListEditPageFormat (ListName: "Subject List"; NumberFormat: DocFormat!)
ListDefinitionStyle (ListName: "Subject List"; Style: ListStyle!)
ListDefinitionEditBox (ListName: "Subject List"; BoxName: FigureBox;)
MarkListUpdateList (ListName: "Subject List")
ListDefinition (ListName: "Subject List")
```

**See Also**

♦ ListDefinition

---

---

## ListDefinitionDelete

---

**Purpose**

ListDefinitionDelete deletes a list definition in the Define List dialog box.

**Syntax**

ListDefinitionDelete (ListName: *Character Expression*)

**Parameters***ListName*

Character Expression: Names a list definition to delete.

**Route**

Tools, List, Define, select a list definition, Delete.

**Example**

To delete a list definition named Old List, the command is:

```
ListDefinitionDelete (ListName: "Old List")
```



---

## ListDefinitionEditBox

---

**Purpose** ListDefinitionEditBox changes the type of graphics box the specified list includes when generated.

**Syntax** ListDefinitionEditBox (Name: *Character Expression*; BoxName: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Name* Character Expression: Names a list definition to edit.

*BoxName* Numeric Equivalent or Enumerated Type: Specifies which type of graphics box to include in the list definition.

- 0 FigureBox!
- 1 TableBox!
- 2 TextBox!
- 3 UserBox!
- 4 EquationBox!
- 5 ButtonBox!
- 6 WatermarkImageBox!
- 7 InlineEquationBox!
- 127 NoBox!

**Route** Tools, List, Define, select a list definition, Edit, Auto Reference Box Captions, select a graphics box type.

**Example** To change a list definition named List 1 to include figure boxes, the command is:

```
ListDefinitionEditBox (Name: "List 1"; BoxName: FigureBox!)
```

---

## ListDefinitionMode

---

**Purpose** ListDefinitionMode changes the numbering format in a list definition.

**Syntax** ListDefinitionMode (ListName: *Character Expression*; Numbering: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*ListName* Character Expression: Names a list definition to edit.

## Numbering

Numeric Equivalent or Enumerated Type: Specifies which numbering format a list definition uses.

- 0 DefNone!
- 1 DefFollows!
- 2 DefParentheses!
- 3 DefFlushRight!
- 4 DefLeader!

---

**Route** Tools, List, Define, select a list definition, Edit, select an option from the Text.....# pop-up list.

---

**Example** To set the numbering format to flush right in a list definition named List 2, the command is:

ListDefinitionMode (ListName: "List 2"; Numbering: DefFlushRight!)

---

## ListDefinitionRename

---

**Purpose** ListDefinitionRename renames a list definition.

---

**Syntax** ListDefinitionRename (OldName: *Character Expression*; NewName: *Character Expression*)

---

### Parameters

*OldName* Character Expression: Specifies a list definition to rename.

*NewName* Character Expression: Specifies the new name of a list definition.

---

**Route** Tools, List, Define, select a list definition, Edit, type a new name.

---

**Example** To rename a list definition named List 1 to List One, the command is:

ListDefinitionRename (OldName: "List 1"; NewName: "List One")

---

## ListDefinitionRetrieve

---

**Purpose** ListDefinitionRetrieve retrieves one list definition from a document.

---

**Syntax** ListDefinitionRetrieve (Filename: *Character Expression*; ListName: *Character Expression*)

---

**Parameters**

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <i>Filename</i> | Character Expression: Names the document which contains a list to retrieve. |
| <i>ListName</i> | Character Expression: Specifies the list definition to retrieve.            |

---

**Route** Tools, List, Define, Retrieve, specify a document name, mark a list to retrieve.

---

**Example** To retrieve a list definition named List One from a file named MYFILE, the command is:

ListDefinitionRetrieve (Filename: "MYFILE"; ListName: "List One")

---

---

## ListDefinitionStyle

---

**Purpose** ListDefinitionStyle edits a list definition and changes the list style that definition uses.

---

**Syntax** ListDefinitionStyle (ListName: *Character Expression*; Style: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ListName</i> | Character Expression: Specifies a list definition to edit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Style</i>    | Numeric Equivalent or Enumerated Type: Specifies which list style to include in the list definition.<br><ul style="list-style-type: none"><li>0 NormalStyle!</li><li>16 FootnoteNumberInDocumentStyle!</li><li>17 EndnoteNumberInDocumentStyle!</li><li>18 FigureNumberStyle!</li><li>19 TableBoxNumberStyle!</li><li>20 TextBoxNumberStyle!</li><li>21 UserBoxNumberStyle!</li><li>22 EquationNumberStyle!</li><li>23 HypertextStyle!</li><li>24 ToC1Style!</li><li>25 ToC2Style!</li><li>26 ToC3Style!</li><li>27 ToC4Style!</li><li>28 ToC5Style!</li><li>29 Index1Style!</li><li>30 Index2Style!</li><li>31 ListStyle!</li><li>32 ToAStyle!</li><li>33 DocStyle!</li><li>34 BoxTextStyle!</li><li>35 CaptionStyle!</li></ul> |

36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

---

**Route**

Tools, List, Define, select a list definition, Edit, Change Current Style, highlight a style, Select.

---

**Example**

To change the list style to Index 2 of a list definition named List One, the command is:

```
ListDefinitionStyle (ListName: "List One"; Style: Index2Style!)
```

---

**See Also** ♦ ListEditPageFormat

---

## ListEditPageFormat

---

**Purpose** ListEditPageFormat edits the page numbering format of a list definition.

---

**Syntax** ListEditPageFormat (ListName: *Character Expression*; NumberFormat: *Character Expression*)

---

### Parameters

*ListName* Character Expression: Specifies a list definition to edit.

*NumberFormat* Character Expression: Specifies a page numbering style. If there is nothing between the quotation marks in this parameter, the generated list uses the same page numbering style as in the main document.

---

**Route** Tools, List, Define, select a list definition, Edit, Page Numbering, specify a page number format.

---

**Example** To specify a page numbering format to be the same as the main document format in a definition named List One, the command is:

```
ListEditPageFormat (ListName: "List One"; NumberFormat: "")
```

---

**See Also** ♦ ListDefinitionStyle

---

## ListMark

---

**Purpose** ListMark marks selected text to add to a list when a document is generated.

IMPORTANT: If no text is selected, this command ends the macro or redirects execution to the LABEL specified by ONERROR.

---

**Syntax** ListMark (ListName: *Character Expression*)

---

### Parameters

*ListName* Character Expression: Names a new list.

---

**Route** Tools, List, Define, specify a list name, Mark.

---

**Example**

To mark selected text to add to a list named List One, the command is:

ListMark (ListName: "List One")

---

## MacroAssignDlg

---

**Purpose**

MacroAssignDlg displays the Assign Macro to Menu dialog box.

**Syntax**

MacroAssignDlg ()  
Tools, Macro, Menu.

---

## MacroCommandInserter

---

**Purpose**

MacroCommandInserter opens or closes the WordPerfect Macro Command Inserter.

**Syntax**

MacroCommandInserter (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type: Turns the WordPerfect Macro Command Inserter on or off.

0 Off!  
1 On!

To turn on the WordPerfect Macro Command Inserter: Tools, Macro, Macro Bar, Command Inserter. To turn it off: click Close.

To turn on the WordPerfect Macro Command Inserter, the command is:

MacroCommandInserter (State: On!)

---

## MacroEditControlBar

---

**Purpose**

MacroEditControlBar displays the Macros feature bar.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

---

---

**Syntax** MacroEditControlBar ()

---

## MacroMenuAdd

---

**Purpose** MacroMenuAdd adds a macro to the macro submenu.

---

**Syntax** MacroMenuAdd (UpdateBIF: *Numeric Equivalent* or *Enumerated Type*; MenuPos: *Numeric Expression*; Location: *Numeric Equivalent* or *Enumerated Type*; Filename: *Character Expression*; TemplateName: *Character Expression*)

---

### Parameters

*UpdateBIF* Numeric Equivalent or Enumerated Type: Specifies whether to save a macro submenu addition.

- 0 No!
- 1 Yes!

*MenuPos* Numeric Expression: Specifies a menu position for an added macro.

*Location* Numeric Equivalent or Enumerated Type: Specifies the location of an added macro.

- 1 FileOnDisk!
- 4 CurrentTemplate!
- 5 StandardTemplate!

*Filename* Character Expression (optional): Specifies a file containing a macro to add.

*TemplateName* Character Expression (optional): Specifies the template containing a macro to add.

---

**Route** Tools, Macro, Menu, Insert, specify a macro name, Select.

---

**Example** To add a macro named CHAIN.WCM in the standard template, to the first position in macro menu, the command is:

```
MacroMenuAdd (UpdateBIF: Yes!; MenuPos: 1; Location: FileOnDisk!; Filename:
"c:\wpwin60\macros\chain.wcm")
```

---

# MacroMenuDelete

---

|                   |                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | MacroMenuDelete deletes a macro from a menu.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | MacroMenuDelete (UpdateBIF: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , MenuPos: <i>Numeric Expression</i> )                               |
| <b>Parameters</b> |                                                                                                                                                      |
| <i>UpdateBIF</i>  | Numeric Expression: Specifies whether to update BIF.<br><br>0 No!<br>1 Yes!                                                                          |
| <i>MenuPos</i>    | Numeric Expression: Specifies a macro's menu position.                                                                                               |
| <b>Example</b>    | To delete a macro from the third position on a menu, the command is:<br><br>MacroMenuDelete (UpdateBIF: No!; MenuPos: 3)                             |

---

# MacroPause

---

|                |                                                                                                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MacroPause stops a macro until the user presses Enter or selects Pause on the Macro menu.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | MacroPause ()                                                                                                                                                                                     |

---

# MacroStatusPrompt

---

|                |                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MacroStatusPrompt specifies whether to display the macro status prompt line.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | MacroStatusPrompt(State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Prompt: <i>Character Expression</i> )                                                                 |



---

**Parameters**

|                |                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------|
| <i>State</i>   | Numeric Equivalent or Enumerated Type:<br>0 Off!<br>1 On!                                                         |
| <i>Prompt</i>  | Character Expression: Specifies a prompt message.                                                                 |
| <b>Example</b> | To display the prompt Please Wait..., the command is:<br>MacroStatusPrompt (State: On!; Prompt: "Please Wait...") |

---

---

**MailDlg**

|                |                                                                                                                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MailDlg determines if WP Office is running; loads WP Office if it is not; displays the WP Office Mail view; attaches a copy of the active WordPerfect document to Mail; and writes "<WP Attachment Enclosed>" in the Message text box. |
| <b>Syntax</b>  | MailDlg ()                                                                                                                                                                                                                             |
| <b>Route</b>   | File, WP Mail.                                                                                                                                                                                                                         |

---

---

**MarginAdjustLeft**

|                    |                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | MarginAdjustLeft moves the left margin relative to the current margin setting.                          |
| <b>Syntax</b>      | MarginAdjustLeft (AmtToAdjust: <i>Measurement Expression</i> )                                          |
| <b>Parameters</b>  |                                                                                                         |
| <i>AmtToAdjust</i> | Measurement Expression: Specifies a positive or negative value to increase or decrease the left margin. |
| <b>Route</b>       | Layout, Paragraph, Format, specify a measurement in the Left Margin Adjustment entry box.               |
| <b>Example</b>     | To move the left margin .5 inches to the right, the command is:<br>MarginAdjustLeft (AmtToAdjust: .5")  |

---

---

**See Also** ♦ MarginLeft ♦ ?LeftMarginAdj

---

## MarginAdjustRight

---

**Purpose** MarginAdjustRight moves the right margin relative to the current margin setting.

---

**Syntax** MarginAdjustRight (AmtToAdjust: *Measurement Expression*)

---

**Parameters**

*AmtToAdjust* Measurement Expression: Specifies a positive or negative value to increase or decrease the right margin.

---

**Route** Layout, Paragraph, Format, Right Margin Adjustment, type a value.

---

**Example** To move the right margin .5 inches to the left, the command is:  
MarginAdjustRight (AmtToAdjust: -.5")

---

**See Also** ♦ MarginRight ♦ ?RightMarginAdj

---

## MarginBottom

---

**Purpose** MarginBottom sets the bottom margin.

---

**Syntax** MarginBottom (MarginWidth: *Measurement Expression*)

---

**Parameters**

*MarginWidth* Measurement Expression: Specifies a bottom margin.

---

**Route** Layout, Margins, Bottom Margin, specify a measurement.

---

**Example** To set the bottom margin 1 inch from the bottom edge of the page, the command is:  
MarginBottom (MarginWidth: 1")

---

**See Also** ♦ ?MarginBottom

---

## MarginLeft

|                    |                                                                                                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | MarginLeft sets the left margin.                                                                                           |
| <b>Syntax</b>      | MarginLeft (MarginWidth: <i>Measurement Expression</i> )                                                                   |
| <b>Parameters</b>  |                                                                                                                            |
| <i>MarginWidth</i> | Measurement Expression: Specifies a left margin.                                                                           |
| <b>Route</b>       | Layout, Margins, Left Margin, specify a measurement.                                                                       |
| <b>Example</b>     | To set the left margin 1 inch from the left edge of the page, the command is:<br><code>MarginLeft (MarginWidth: 1")</code> |
| <b>See Also</b>    | ♦ MarginAdjustLeft ♦ ?MarginLeft                                                                                           |

---

## MarginReleaseInsert

|                |                                                                         |
|----------------|-------------------------------------------------------------------------|
| <b>Purpose</b> | MarginReleaseInsert moves the insertion point one tab stop to the left. |
| <b>Syntax</b>  | MarginReleaseInsert ()                                                  |
| <b>Route</b>   | Press Ctrl+Shift+Tab.                                                   |

---

## MarginRight

|                    |                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | MarginRight sets the right margin.                                                                                            |
| <b>Syntax</b>      | MarginRight (MarginWidth: <i>Measurement Expression</i> )                                                                     |
| <b>Parameters</b>  |                                                                                                                               |
| <i>MarginWidth</i> | Measurement Expression: Specifies a right margin.                                                                             |
| <b>Route</b>       | Layout, Margins, Right Margin, specify a measurement.                                                                         |
| <b>Example</b>     | To set the right margin 1 inch from the right edge of the page, the command is:<br><code>MarginRight (MarginWidth: 1")</code> |

---

**See Also** ♦ MarginAdjustRight ♦ ?MarginRight

---

## MarginTop

---

**Purpose** MarginTop sets the top margin.

---

**Syntax** MarginTop (MarginWidth: *Measurement Expression*)

---

**Parameters**

*MarginWidth* Measurement Expression: Specifies a top margin.

---

**Route** Layout, Margins, Top Margin, specify a measurement.

---

**Example** To set the top margin 1.5 inch from the top edge of the page, the command is:  
MarginTop (MarginWidth: 1.5")

---

**See Also** ♦ ?MarginTop

---

## MarkCrossRef

---

**Purpose** MarkCrossRef marks a cross reference.

---

**Syntax** MarkCrossRef ()

---

**Route** Tools, Cross Reference.

---

**See Also** ♦ MarkCrossRefUpdateLink

---

## MarkCrossRefUpdateList

---

**Purpose** MarkCrossRefUpdateList creates a cross reference to a specified target name.

---

**Syntax** MarkCrossRefUpdateList (Target: *Character Expression*)

---

**Parameters**

*Target* Character Expression: Specifies a target name.

**Route** Tools, Cross Reference, Mark Target, Mark Reference.

**Example** To mark the target name Archaeology, the command is:

MarkCrossRefUpdateList (Target: "Archaeology")

**See Also** ♦ MarkCrossRef

---

## MarkIndex

**Purpose** MarkIndex marks a word for the index.

**Syntax** MarkIndex ()

**Route** Tools, Index.

**See Also** ♦ MarkListUpdateIndex

---

## MarkIndexUpdateList

**Purpose** MarkIndexUpdateList creates a reference to a specified heading and subheading in the index.

**Syntax** MarkIndexUpdateList (Heading: *Character Expression*; SubHeading: *Character Expression*)

**Parameters**

*Heading* Character Expression: Specifies an index heading.

*SubHeading* Character Expression: Specifies a subheading.

**Route** Tools, Index, specify a heading and subheading, Mark.

**Example** To mark a page for the index under the heading Middle East and subheading Lebanon, the command is:

MarkIndexUpdateList (Heading: "Middle East"; SubHeading: "Lebanon")

---

**See Also** ♦ MarkIndex

---

## MarkList

---

**Purpose** MarkList marks an item for a list.

---

**Syntax** MarkList ()

---

**Route** Tools, List.

---

**See Also** ♦ MarkListUpdateList

---

## MarkListUpdateList

---

**Purpose** MarkListUpdateList creates a reference to a specified list name.

---

**Syntax** MarkListUpdateList (ListName: *Character Expression*)

---

**Parameters**

*ListName* Character Expression: Specifies a list name.

---

**Route** Tools, List, select a name from the List pop-up list, Mark button.

---

**Example** To mark the list name Cocker Spaniels, the command is:

MarkListUpdateList (ListName: "Cocker Spaniels")

---

**See Also** ♦ MarkList

---

## MarkTableOfAuthorities

---

**Purpose** MarkTableOfAuthorities marks an item for a table of authorities.

---

**Syntax** MarkTableOfAuthorities ()

---

|              |                              |
|--------------|------------------------------|
| <b>Route</b> | Tools, Table of Authorities. |
|--------------|------------------------------|

---

## MarkTableOfContents

---

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Purpose</b> | MarkTableOfContents marks an item for a table of contents. |
|----------------|------------------------------------------------------------|

---

|               |                        |
|---------------|------------------------|
| <b>Syntax</b> | MarkTableOfContents () |
|---------------|------------------------|

---

|              |                           |
|--------------|---------------------------|
| <b>Route</b> | Tools, Table of Contents. |
|--------------|---------------------------|

---

## MarkToAUpdateList

---

|                |                                                                                          |
|----------------|------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MarkToAUpdateList creates a reference to a specified citation in a table of authorities. |
|----------------|------------------------------------------------------------------------------------------|

---

|               |                                                            |
|---------------|------------------------------------------------------------|
| <b>Syntax</b> | MarkToAUpdateList (ListName: <i>Character Expression</i> ) |
|---------------|------------------------------------------------------------|

---

|                   |  |
|-------------------|--|
| <b>Parameters</b> |  |
|-------------------|--|

---

|                  |                                                                        |
|------------------|------------------------------------------------------------------------|
| <i>ShortForm</i> | Character Expression: Specifies a list name in a table of authorities. |
|------------------|------------------------------------------------------------------------|

---

|              |                                                                                    |
|--------------|------------------------------------------------------------------------------------|
| <b>Route</b> | Tools, Table of Authorities, select a name from the List pop-up list, Mark button. |
|--------------|------------------------------------------------------------------------------------|

---

|                |                                                              |
|----------------|--------------------------------------------------------------|
| <b>Example</b> | To specify the list name Barclay v. Florida, the command is: |
|----------------|--------------------------------------------------------------|

MarkToAUpdateList (ShortForm: "Barclay v. Florida")

---

## MasterDocCondense

---

|                |                                                                                             |
|----------------|---------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MasterDocCondense specifies subdocuments to condense and specifies whether to save changes. |
|----------------|---------------------------------------------------------------------------------------------|

---

|               |                                                                                                                          |
|---------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | MasterDocCondense (Filename: <i>Character Expression</i> ; Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|--------------------------------------------------------------------------------------------------------------------------|

---

## Parameters

|                 |                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Filename</i> | Character Expression: Specifies a subdocument to condense.                                                                                               |
| <i>Action</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to save changes to a subdocument when the subdocument is condensed.<br>0 SkipSub!<br>2 SaveSub! |

---

|              |                                                               |
|--------------|---------------------------------------------------------------|
| <b>Route</b> | File, Master Document, Condense Master, select a subdocument. |
|--------------|---------------------------------------------------------------|

---

|                |                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To condense and not save changes in a subdocument named C:\WPWIN60\SUB1.DOC, the command is:<br><br>MasterDocCondense (Filename: "C:\WPWIN60\SUB1.DOC"; Action: SkipSub!) |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## MasterDocCondenseDlg

---

|                |                                                                          |
|----------------|--------------------------------------------------------------------------|
| <b>Purpose</b> | MasterDocCondenseDlg displays the Condense/Save Subdocuments dialog box. |
|----------------|--------------------------------------------------------------------------|

---

|               |                         |
|---------------|-------------------------|
| <b>Syntax</b> | MasterDocCondenseDlg () |
|---------------|-------------------------|

---

|              |                                         |
|--------------|-----------------------------------------|
| <b>Route</b> | File, Master Document, Condense Master. |
|--------------|-----------------------------------------|

---

|                 |                     |
|-----------------|---------------------|
| <b>See Also</b> | ♦ MasterDocCondense |
|-----------------|---------------------|

---

## MasterDocExpand

---

|                |                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MasterDocExpand starts the expansion of a subdocument.<br><br>IMPORTANT: This command is used in combination with SubDocInclude, which specifies a subdocument to expand. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                    |
|---------------|--------------------|
| <b>Syntax</b> | MasterDocExpand () |
|---------------|--------------------|

---

|              |                                                             |
|--------------|-------------------------------------------------------------|
| <b>Route</b> | File, Master Document, Expand Master, select a subdocument. |
|--------------|-------------------------------------------------------------|

---

|                |                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To expand a subdocument named SUB1.DOC, the command sequence is:<br><br>MasterDocExpand ()<br>SubDocInclude (Filename: "SUB1.DOC") |
|----------------|------------------------------------------------------------------------------------------------------------------------------------|



---

## MasterDocSubdocDlg

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| <b>Purpose</b> | MasterDocSubdocDlg displays the Include Subdocument dialog box. |
| <b>Syntax</b>  | MasterDocSubdocDlg ()                                           |
| <b>Route</b>   | File, Master Document, Subdocument.                             |

---

## MatchExtendSelection

|                |                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MatchExtendSelection specifies whether Find selects the text between the insertion point and the matched text. |
| <b>Syntax</b>  | MatchExtendSelection ()                                                                                        |
| <b>Route</b>   | Edit, Find, Action, Extend Selection.                                                                          |

---

## MatchLimit

|                |                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MatchLimit limits a find and replace search to a specified number.                                               |
| <b>Syntax</b>  | MatchLimit (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , LimitValue: <i>Numeric Expression</i> ) |

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to limit a find and replace search.

0 No!  
1 Yes!

*LimitValue* Numeric Expression: Specifies the number of find and replace searches.

|              |                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Edit, Replace, type Find and Replace With strings, Options, Limit Number of Changes, enter a value, Replace All. |
|--------------|------------------------------------------------------------------------------------------------------------------|

|                |                                                         |
|----------------|---------------------------------------------------------|
| <b>Example</b> | To limit find and replace to 2 matches, the command is: |
|----------------|---------------------------------------------------------|

MatchLimit (State: Yes!; LimitValue: 2)

---

## MatchPositionAfter

|                |                                                                                      |
|----------------|--------------------------------------------------------------------------------------|
| <b>Purpose</b> | MatchPositionAfter places the insertion point after a search string when using Find. |
| <b>Syntax</b>  | MatchPositionAfter ()                                                                |
| <b>Route</b>   | Edit, Find, Action, Position After, enter search string.                             |

---

## MatchPositionBefore

|                |                                                                                          |
|----------------|------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MatchPositionBefore places the insertion point before the search string when using Find. |
| <b>Syntax</b>  | MatchPositionBefore ()                                                                   |
| <b>Route</b>   | Edit, Find, Action, Position Before, enter search string.                                |

---

## MatchSelection

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| <b>Purpose</b> | MatchSelection specifies whether Find selects the matched text. |
| <b>Syntax</b>  | MatchSelection ()                                               |
| <b>Route</b>   | Edit, Find, Action, Select Match.                               |

---

## MatchWithAttributes

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| <b>Purpose</b> | MatchWithAttributes specifies whether Find matches only text with selected attributes. |
| <b>Syntax</b>  | MatchWithAttributes (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )      |

### Parameters

|              |                                                           |
|--------------|-----------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type:<br>0 No!<br>1 Yes! |
|--------------|-----------------------------------------------------------|

|                 |                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | No! = Edit, Find, Match, Font, select Text Only. Yes! = Edit, Find, Match, Font, select one or more attributes. |
| <b>Example</b>  | To find text with selected attributes, the command is:<br>MatchWithAttributes (State: Yes!)                     |
| <b>See Also</b> | ♦ MatchWithFont ♦ MatchWithFontSize                                                                             |

---

## MatchWithFont

|                |                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MatchWithFont specifies whether Find matches only text with selected font attributes.                                                                                                                                                                                                                                                                                                                            |
| <b>Syntax</b>  | MatchWithFont (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Typeface: <i>Character Expression</i> ; Family: <i>Numeric Expression</i> ; Attributes: <i>Numeric Expression</i> ; Weight: <i>Numeric Expression</i> ; Width: <i>Numeric Expression</i> ; Source: <i>Numeric Expression</i> ; Type: <i>Numeric Expression</i> ; CharacterSet: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

### Parameters

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>State</i>    | Numeric Equivalent or Enumerated Type:<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>Typeface</i> | Character Expression: Specifies a typeface name.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Family</i>   | Numeric Equivalent or Enumerated Type:<br>256 FamilyITCBenguiat!<br>2304 FamilyTimesRoman!<br>4352 FamilyITCGaramond!<br>6400 FamilyCenturySchoolbook!<br>8448 FamilyBodon!<br>10496 FamilyITCLubalinGraph!<br>12544 FamilyAlbertus!<br>14592 FamilyCourier!<br>16640 FamilyHelvetica!<br>18688 FamilyITCRonda!<br>20736 FamilyMicrostyle!<br>22784 FamilyOptima!<br>24832 FamilyLetterGothic!<br>26880 FamilyBrush!<br>28928 FamilyParkAvenue!<br>30976 FamilySignetRoundhand!<br>33024 FamilyTekton! |

35072 FamilyUncial!  
37120 FamilyOldEnglish!  
39168 FamilyITCZapfDingbats!  
41216 FamilyBroadway!  
43264 FamilyComputer!  
45312 FamilyPTBarnum!  
47360 FamilyPonderosa!  
49408 FamilyMadrone!  
51456 FamilyHobo!  
53504 FamilyRevue!  
55552 FamilyAestheticOrnamented!

*Attributes*

Numeric Equivalent or Enumerated Type:

0 FontMatchNormal!  
1 FontMatchItalic!  
2 FontMatchOutline!  
4 FontMatchShadow!  
8 FontMatchSmallCaps!

*Weight*

Numeric Equivalent or Enumerated Type:

0 WeightUnknown!  
1 WeightUltraThin!  
8 WeightUltraLight!  
16 WeightExtraThin!  
24 WeightThin!  
32 WeightExtraLight!  
48 WeightLight!  
64 WeightDemiLight!  
80 WeightSemiLight!  
88 WeightBook!  
88 WeightRegular!  
104 WeightRoman!  
112 WeightMedium!  
128 WeightDemiBold!  
136 WeightSemiBold!  
144 WeightBold!  
160 WeightExtraBold!  
176 WeightHeavy!  
192 WeightExtraHeavy!  
208 WeightBlack!  
224 WeightExtraBlack!  
240 WeightUltraHeavy!  
248 WeightUltraBlack!  
254 WeightMaximum!

*Width* Numeric Equivalent or Enumerated Type:  
0 WidthUnknown!  
16 WidthUltraCondensed!  
32 WidthUltraCompressed!  
48 WidthCompressed!  
64 WidthExtraCondensed!  
80 WidthCondensed!  
96 WidthSemiCondensed!  
112 WidthNormal!  
128 WidthSemiExpanded!  
144 WidthExpanded!  
160 WidthExtraExpanded!  
176 WidthUltraExpanded!  
192 WidthDoubleWide!  
224 WidthTripleWide!

*Source* Numeric Equivalent or Enumerated Type:  
16 PRSFile!  
20 DRSFile!

*Type* Numeric Equivalent or Enumerated Type:  
134 Speedo!  
135 Intellifont!  
138 Type1!  
139 TrueType!

*CharacterSet* Numeric Equivalent or Enumerated Type:  
0 FontMatchASCII!  
1 FontMatchMultinational1!  
2 FontMatchMultinational2!  
3 FontMatchBoxDrawing!  
4 FontMatchTypographicSymbols!  
5 FontMatchIconicSymbols!  
6 FontMatchMath!  
7 FontMatchMathExtension!  
8 FontMatchGreek!  
9 FontMatchHebrew!  
10 FontMatchCyrillic!  
11 FontMatchJapaneseKana!  
12 FontMatchUserDefined!  
13 FontMatchArabic!  
14 FontMatchArabicScript!

---

**Route**

Edit, Find, Match, Font, select a font from the font pop-up list, select options.

---

**Example**

To find text with selected font attributes, the command is:

MatchWithFont (State: Yes!; Name: "Courier New Bold"; Family: FamilyCourier!; Attributes: FontMatchNormal!; Weight: WeightExtraBold!; Width: WidthUnknown!; Source: DRSFile!; Type: TrueType!; CharacterSet: FontMatchASCII!)

---

**See Also**

♦ MatchWithFontSize ♦ MatchWithAttributes

---

---

## MatchWithFontSize

---

**Purpose**

MatchWithFontSize specifies whether Find matches only text with the selected font size.

---

**Syntax**

MatchWithFontSize (State: *Numeric Equivalent* or *Enumerated Type*; FontSize: *Numeric Expression*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type:

0 No!  
1 Yes!

*FontSize*

Numeric Expression: Specifies a font height in points.

---

**Route**

No! = Edit, Find, Match, Font, select Text Only. Yes! = Edit, Find, Match, Font, select attributes.

---

**Example**

To find text with a 14 point font size, the command is:

MatchWithFontSize (State: Yes!; FontSize: 14p)

---

**See Also**

♦ MatchWithFont ♦ MatchWithAttributes

---

---

## MenuAddItem

---

**Purpose**

MenuAddItem adds an item to a WordPerfect menu or personalized submenus.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** MenuAddItem (MenuBarName: *Character Expression*; Template: *Numeric Equivalent or Enumerated Type*; MenuName: *Character Expression*; ItemName: *Character Expression*; Offset: *Numeric Expression*; Type: *Numeric Equivalent or Enumerated Type*; MacroName: *Character Expression*; ItemText: *Character Expression*; Prompt: *Character Expression*; HelpFile: *Character Expression*; HelpId: *Numeric Expression*; Script: *Character Expression*)

---

**Parameters**

*MenuBarName* Character Expression: Specifies the menu bar name.

*Template* Numeric Equivalent or Enumerated Type: Specifies the location of the template to which the menu is saved.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

*MenuName* Character Expression: Specifies the menu to add an item to.

*ItemName* Character Expression: Specifies a descriptive name of a menu item to add.

*Offset* Numeric Expression: Specifies the location of a menu item in a list of items. Values range from 1 to 255.

*Type* Numeric Equivalent or Enumerated Type: Specifies the type of menu item to create.

- 0 Macro!
- 1 Separator!
- 2 Submenu!
- 3 Feature!
- 4 Script!
- 5 Launch!
- 6 Dyna!

*MacroName* Character Expression: Specifies the name of a macro that plays when the corresponding menu item is chosen.

*ItemText* Character Expression: Specifies menu text for the created menu item.

*Prompt* Character Expression: Specifies text to display on the status line when the corresponding menu item is selected.

*HelpFile* Character Expression: Specifies the name of a Help file to use if Help is the menu item.

*HelpId* Numeric Expression: Specifies the ID number of the Help file in the HelpFile parameter.

*Script* Character Expression: Specifies the text to type when the script menu item is chosen.

---

**Example**

To add an macro named C:\WPWIN60\MACROS\BOLD.WCM to the Tools menu on the WPWIN 6.0 menu bar, the command is:

```
MenuAddItem (MenuBarName: "<WPWIN 6.0 Menu>"; Template: CurrentDoc!; MenuName:
"ToolsMenu"; ItemName: ""; Offset: 0; Type: Macro!; MacroName:
"C:\WPWIN60\MACROS\BOLD.WCM"; ItemText: "Bold Macro"; Prompt: "Apply bold attribute to the
current word")
```

---

**See Also**

♦ MenuDeleteItem

---

---

## MenuDeleteItem

**Purpose**

MenuDeleteItem deletes an item from a specified menu.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

MenuDeleteItem (MenuName: *Character Expression*; ItemName: *Character Expression*)

---

**Parameters**

*MenuName*

Character Expression: Specifies the name of the menu from which to delete an item.

*ItemName*

Character Expression: Specifies the name of a menu item to delete.

---

**Example**

To delete a macro from the Macro menu, the command is:

```
MenuDeleteItem (MenuName: "Macro"; ItemName: "Test Macro")
```

---

**See Also**

♦ MenuAddItem

---

---

## MenuSelect

**Purpose**

MenuSelect selects a menu.

---

**Syntax**

MenuSelect (MenuName: *Character Expression*)

---

**Parameters**

*MenuName*

Character Expression: Specifies a menu name.

---



|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| <b>Route</b>    | File, Preferences, Menu Bar Icon, select a menu.                                  |
| <b>Example</b>  | To select the menu Macros, the command is:<br><br>MenuSelect (MenuName: "Macros") |
| <b>See Also</b> | ♦ MenuSelectDlg                                                                   |

---

## MenuSelectDlg

|                 |                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | MenuSelectDlg displays the Menu Bar Preferences dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | MenuSelectDlg ()                                                                                                                                                    |
| <b>See Also</b> | ♦ MenuSelect                                                                                                                                                        |

---

## MergeBlankField

|                   |                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | MergeBlankField specifies whether to remove a blank field. If blank fields are removed, this command shifts remaining fields up one line. |
| <b>Syntax</b>     | MergeBlankField (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                             |
| <b>Parameters</b> |                                                                                                                                           |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to remove a blank field.<br><br>0 Leave!<br>1 Remove!                            |
| <b>Route</b>      | Tools, Merge, Merge, Options, select an option from the If Empty Field in Data File pop-up list.                                          |
| <b>Example</b>    | To remove a blank field, the command is:<br><br>MergeBlankField (State: Remove!)                                                          |

---

# MergeCode

---

**Purpose** MergeCode inserts a merge code into the current document.

---

**Syntax** MergeCode (Codes: *Numeric Equivalent* or *Enumerated Type*, String: *Character Expression*)

---

## Parameters

*Codes* Numeric Equivalent or Enumerated Type: Specifies a merge code to insert.

- 0 Beep!
- 1 Break!
- 2 CancelOff!
- 3 CancelOn!
- 4 Continue!
- 5 Date!
- 6 Default!
- 7 DisplayStop!
- 8 Else!
- 9 EndSwitch!
- 10 EndField!
- 11 EndFor!
- 12 EndIf!
- 13 EndRecord!
- 14 EndWhile!
- 15 Next!
- 16 NextRecord!
- 17 PageOff!
- 18 PageOn!
- 19 Print!
- 20 Quit!
- 21 ProcessOn!
- 22 Return!
- 23 ReturnCancel!
- 24 ReturnError!
- 25 Rewrite!
- 26 StepOff!
- 27 StepOn!
- 28 Stop!
- 29 ProcessOff!
- 48 Assign!
- 50 Call!
- 52 Caps!
- 54 Switch!
- 56 CaseOf!
- 58 ChainMacro!
- 60 ChainData!

62 ChainForm!  
64 Char!  
66 Codes!  
68 Comment!  
70 CtoN!  
72 Document!  
74 EmbedMacro!  
76 Field!  
78 FieldNames!  
80 FirstCap!  
82 For!  
84 ForEach!  
86 GetString!  
88 Go!  
90 If!  
92 IfBlank!  
94 IfExists!  
96 IfNotBlank!  
98 Insert!  
100 Keyboard!  
102 Label!  
104 AssignLocal!  
106 Look!  
108 MrgCmnd!  
110 NestData!  
112 NestForm!  
114 NestMacro!  
116 NtoC!  
118 OnCancel!  
120 OnError!  
122 Prompt!  
124 StatusPrompt!  
126 StrLen!  
128 StrPos!  
130 SubstData!  
132 SubstForm!  
134 SubStr!  
136 System!  
138 ToLower!  
140 ToUpper!  
142 Variable!  
144 Wait!  
146 While!  
148 PostNet!

|                |                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>String</i>  | Character Expression: Specifies field parameters if needed.                                                                                                                        |
| <b>Route</b>   | Select the Merge Codes button on the Merge Feature Bar. The Merge Feature Bar is activated when a merge form file or data file is opened.                                          |
| <b>Example</b> | To insert a FieldNames! code with name, address and telephone fields, the command is:<br><pre>MergeCode (Codes: FieldNames!; String: "name; address; city/state/zip; phone")</pre> |

---

## MergeCodesDisplayEdit

|                   |                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | MergeCodesDisplayEdit hides or shows merge codes, or displays them as markers.                                                                                                                        |
| <b>Syntax</b>     | MergeCodesDisplayEdit (Display: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                                       |
| <i>Display</i>    | Numeric Equivalent or Enumerated Type: Specifies whether to show merge codes or display them as markers.<br><ul style="list-style-type: none"><li>0 Hide!</li><li>1 Show!</li><li>2 Marker!</li></ul> |
| <b>Route</b>      | Select the Options button on the Merge feature bar. The Merge feature bar is activated when a merge form file or data file is opened.                                                                 |
| <b>Example</b>    | To hide merge codes, the command is:<br><pre>MergeCodesDisplayEdit (Display: Hide!)</pre>                                                                                                             |
| <b>See Also</b>   | ♦ MergeCodesDisplayRun                                                                                                                                                                                |

---

## MergeCodesDisplayRun

|                |                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeCodesDisplayRun hides or shows merge codes, or displays them as markers in a merged document. |
| <b>Syntax</b>  | MergeCodesDisplayRun (Display: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )               |

---

**Parameters**

*Display* Numeric Equivalent or Enumerated Type: Specifies whether to show merge codes or display them as markers.

- 0 Hide!
- 1 Show!
- 2 Marker!

---

**Route** Tools, Merge, Merge, Options, select an option from the Display Options pop-up list.

---

**Example** To show merge codes, the command is:  

```
MergeCodesDisplayRun (Display: Show!)
```

---

**See Also** ♦ MergeCodesDisplayEdit

---

## MergeCodesDlg

---

**Purpose** MergeCodesDlg displays the Insert Merge Codes dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** MergeCodesDlg ()

---

## MergeCondition

---

**Purpose** MergeCondition defines a condition for selecting records from a data file.

---

**Syntax** MergeCondition (Condition1: *Character Expression*; Condition2: *Character Expression*; ... Condition15: *Character Expression*)

---

**Parameters**

*Condition1*,  
*Condition2*, ...  
*Condition15*

Character Expression: Specifies a condition for record selection.

---

**Route** Tools, Merge, Merge, specify a form file and a data file, Select Records, Specify Conditions, specify fields and conditions.

---

**Example**

To select all records that match the following condition set: 1) COMPANY = WordPerfect; (2) TITLE = Regional Manager; (3) STATE = CA, the command is:

MergeCondition (Condition1: "COMPANY"; Condition2: "TITLE"; Condition3: "STATE"; Condition4: "WordPerfect"; Condition5: "Regional Manager"; Condition6: "CA")

---

## MergeDataFile

---

**Purpose**

MergeDataFile specifies the data file to associate with a specified form file.

**Syntax**

MergeDataFile (DataFile: *Character Expression*)

**Parameters***DataFile*

Character Expression (optional): Specifies the name of a data file to associate with a specified form file.

**Route**

Tools, Merge, Form specify a data file to associate.

**Example**

To associate a data file named MYLIST with the current form file, the command is:

MergeDataFile (DataFile: "MYLIST")

---

## MergeDlg

---

**Purpose**

MergeDlg displays the Merge dialog box.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax**

MergeDlg ()

---

## MergeEnvelope

---

**Purpose**

MergeEnvelope specifies whether to create an envelope for merged records.

**Syntax**

MergeEnvelope (State: *Numeric Equivalent* or *Enumerated Type*)

---

---

## Parameters

|              |                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether to create an envelope for merged records.<br><br>0 Off!<br>1 On! |
|--------------|---------------------------------------------------------------------------------------------------------------------------|

---

|              |                                               |
|--------------|-----------------------------------------------|
| <b>Route</b> | Tools, Merge, Merge button, Envelopes button. |
|--------------|-----------------------------------------------|

---

|                |                                                                               |
|----------------|-------------------------------------------------------------------------------|
| <b>Example</b> | To create a merge envelope, the command is:<br><br>MergeEnvelope (State: On!) |
|----------------|-------------------------------------------------------------------------------|

---

## MergeEnvelopeDefinitionBegin

---

|                |                                                                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeEnvelopeDefinitionBegin is the opening command of the MergeEnvelopeDefinitionBegin-MergeEnvelopeDefinitionEnd statement; this statement contains address information.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                                 |
|---------------|---------------------------------|
| <b>Syntax</b> | MergeEnvelopeDefinitionBegin () |
|---------------|---------------------------------|

---

|                 |                 |
|-----------------|-----------------|
| <b>See Also</b> | ♦ MergeEnvelope |
|-----------------|-----------------|

---

## MergeEnvelopeDefinitionEnd

---

|                |                                                                                                                                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeEnvelopeDefinitionEnd is the closing command of a MergeEnvelopeDefinitionBegin-MergeEnvelopeDefinitionEnd statement; this statement contains address information.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                               |
|---------------|-------------------------------|
| <b>Syntax</b> | MergeEnvelopeDefinitionEnd () |
|---------------|-------------------------------|

---

|                 |                 |
|-----------------|-----------------|
| <b>See Also</b> | ♦ MergeEnvelope |
|-----------------|-----------------|

---

## MergeFileAssociate

---

|                   |                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | MergeFileAssociate associates primary and secondary files.                                                                        |
| <b>Syntax</b>     | MergeFileAssociate (FormFile: <i>Character Expression</i> ; DataFile: <i>Character Expression</i> )                               |
| <b>Parameters</b> |                                                                                                                                   |
| <i>FormFile</i>   | Character Expression: Specifies a form file.                                                                                      |
| <i>DataFile</i>   | Character Expression: Specifies a data file.                                                                                      |
| <b>Route</b>      | Tools, Merge, Merge, specify a form file and a data file.                                                                         |
| <b>Example</b>    | To associate DOC1 and DOC2, the command is:<br><pre>MergeFileAssociate (FormFile: "C:\WP60\DOC1"; DataFile: "C:\WP60\DOC2")</pre> |

---

## MergeFilesDlg

---

|                |                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeFilesDlg displays the Perform Merge dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | MergeFilesDlg()                                                                                                                                              |

---

## MergeFileType

---

|                   |                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | MergeFileType specifies a file type to merge.                                               |
| <b>Syntax</b>     | MergeFileType (FileType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )              |
| <b>Parameters</b> |                                                                                             |
| <i>FileType</i>   | Numeric Equivalent or Enumerated Type:<br>0 None!<br>1 Form!<br>2 TableData!<br>4 TextData! |



---

|                |                                                                          |
|----------------|--------------------------------------------------------------------------|
| <b>Route</b>   | Tools, Merge, Form or Data.                                              |
| <b>Example</b> | To merge a form file, the command is:<br>MergeFileType (FileType: Form!) |

---

## MergeFormDlg

---

|                |                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeFormDlg displays the Create Merge File dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | MergeFormDlg ()                                                                                                                                             |

---

## MergeMark

---

|                   |                                                                          |
|-------------------|--------------------------------------------------------------------------|
| <b>Purpose</b>    | MergeMark marks a record to include in a merge.                          |
| <b>Syntax</b>     | MergeMark (RecordNum: <i>Numeric Expression</i> )                        |
| <b>Parameters</b> |                                                                          |
| <i>RecordNum</i>  | Numeric Expression: Specifies a record number to mark for merging.       |
| <b>Route</b>      | Tools, Merge, Merge, Select Records, Mark Records, mark records.         |
| <b>Example</b>    | To mark record 5 in a merge, the command is:<br>MergeMark (RecordNum: 5) |

---

## MergePageBreak

---

|                   |                                                                                  |
|-------------------|----------------------------------------------------------------------------------|
| <b>Purpose</b>    | MergePageBreak specifies whether to separate merged documents with a page break. |
| <b>Syntax</b>     | MergePageBreak (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )     |
| <b>Parameters</b> |                                                                                  |

---

*State*

Numeric Equivalent or Enumerated Type:

0 Off!  
1 On!

---

**Route**

Tools, Merge, Merge button, Options button, Separate Each Merged Document with a Page Break.

---

**Example**

To separate merged documents with a page break, the command is:

MergePageBreak (State: On!)

---

## MergePrintText

---

**Purpose**

MergePrintText sends a secondary merge file form to a printer.

---

**Syntax**

MergePrintText ()

---

**Route**

From the Merge feature bar: Options, Print. The Merge feature bar is activated when a merge form file or data file is opened.

---

**See Also**

MergeFileType

---

## MergeRepeat

---

**Purpose**

MergeRepeat sets the number of times to merge each record.

---

**Syntax**

MergeRepeat (NumberToRepeat: *Numeric Expression*)

---

**Parameters**

*NumberToRepeat*

Numeric Expression: Specifies how many times to merge each data record.

---

**Route**

Tools, Merge, Merge, Options, specify a number.

---

**Example**

To merge a data record 3 times, the command is:

MergeRepeat (NumberToRepeat: 3)

# MergeRun

---

|                       |                                                                                                                                                                                                                                                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>        | MergeRun merges a form file and a data file.                                                                                                                                                                                                                                                                                 |
| <b>Syntax</b>         | MergeRun (FormFileType: <i>Numeric Equivalent or Enumerated Type</i> , FormFile: <i>Character Expression</i> ; DataFileType: <i>Numeric Equivalent or Enumerated Type</i> , DataFile: <i>Character Expression</i> ; OutputFileType: <i>Numeric Equivalent or Enumerated Type</i> ; OutputFile: <i>Character Expression</i> ) |
| <b>Parameters</b>     |                                                                                                                                                                                                                                                                                                                              |
| <i>FormFileType</i>   | Numeric Equivalent or Enumerated Type: Specifies a form file type.<br>0 Current!<br>1 Clipboard!<br>2 FormFile!                                                                                                                                                                                                              |
| <i>FormFile</i>       | Character Expression: Specifies a form file.                                                                                                                                                                                                                                                                                 |
| <i>DataFileType</i>   | Numeric Equivalent or Enumerated Type: Specifies a data file type.<br>0 NoData!<br>1 Clipboard!<br>2 DataFile!                                                                                                                                                                                                               |
| <i>DataFile</i>       | Character Expression: Specifies a data file.                                                                                                                                                                                                                                                                                 |
| <i>OutputFileType</i> | Numeric Equivalent or Enumerated Type: Specifies where to send merged output.<br>0 ToCurrentDoc!<br>1 ToNewDoc!<br>2 ToFile!<br>3 ToPrinter!                                                                                                                                                                                 |
| <i>OutputFile</i>     | Character Expression (optional): Specifies the file where to send merge output when ToFile! is specified in the OutputFileType parameter.                                                                                                                                                                                    |
| <b>Route</b>          | Tools, Merge, Merge, specify a form filename, and a data filename, select an option from the Output File pop-up list.                                                                                                                                                                                                        |
| <b>Example</b>        | To merge a form file named C:\WPWIN60\FORM and a data file named C:\WPWIN60\DATA, and to create a new document to contain the output, the command is:<br><br>MergeRun (FormFileType: FormFile!; FormFile: "C:\WPWIN60\FORM"; DataFileType: DataFile!; DataFile: "C:\WPWIN60\DATA"; OutputFileType: ToNewDoc!)                |

---

---

# MergeSelect

---

|                   |                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | MergeSelect identifies records to merge.                                                                                                           |
| <b>Syntax</b>     | MergeSelect (SelectType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Begin: <i>Numeric Expression</i> ; End: <i>Numeric Expression</i> ) |
| <b>Parameters</b> |                                                                                                                                                    |
| <i>SelectType</i> | Numeric Equivalent or Enumerated Type: Specifies records to merge.<br>0 All!<br>1 Marked!<br>2 Range!                                              |
| <i>Begin</i>      | Numeric Expression: Specifies the low end of a range.                                                                                              |
| <i>End</i>        | Numeric Expression: Specifies the high end of a range.                                                                                             |
| <b>Route</b>      | Tools, Merge, Merge, specify files, Select Records, select options.                                                                                |
| <b>Example</b>    | To merge records 10 through 50, the command is:<br><br>MergeSelect (SelectType: Range!; Begin: 10; End: 50)                                        |
| <b>See Also</b>   | MergeMark                                                                                                                                          |

---

# MergeTableDataDlg

---

|                |                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeTableDataDlg displays the Create Data File dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | MergeTableDataDlg ()                                                                                                                                                |

---

# MergeTextDataDlg

---

|                |                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | MergeTextDataDlg displays the Create Merge File dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

**Syntax** MergeTextDataDlg ()

---

## MinusSign

---

**Purpose** MinusSign is the same as pressing the minus sign or hyphen.

**Syntax** MinusSign ()

**Route** Press the minus sign or hyphen key.

---

## MoveModeEnd

---

**Purpose** MoveModeEnd is the closing command for moving text.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** MoveModeEnd ()

---

## NewHFWOccurrence

---

**Purpose** NewHFWOccurrence specifies where to display headers, footers, and watermarks.

**Syntax** NewHFWOccurrence (Occurrence: *Numeric Equivalent* or *Enumerated Type*)

### Parameters

*Occurrence* Numeric Equivalent or Enumerated Type:

- 0 NoPages!
- 1 OddPages!
- 2 EvenPages!
- 3 AllPages!

**Route** Layout, Header/Footer or Watermark, Create or Edit, Placement button, select an option.

---

---

**Example**

To display a header on every page, the command is:

NewHFWOccurrence (Occurrence: AllPages!)

---

## NumberingOptionDlg

---

**Purpose**

NumberingOptionDlg displays the Page Numbering Options dialog box.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

NumberingOptionDlg ()

---

## OLEAcceptLinks

---

**Purpose**

OLEAcceptLinks accepts a link updated or initialized between server and client applications.

---

**Syntax**

OLEAcceptLinks ()

---

**Route**

Edit, Links, select a link, Update Now.

---

**See Also**

♦ OLEInitializeLinks ♦ OLEUpdateLink

---

## OLECancelLink

---

**Purpose**

OLECancelLink cancels a link to an object.

---

**Syntax**

OLECancelLink (LinkName: *Character Expression*)

---

**Parameters**

*LinkName*

Character Expression: Specifies the name of the link to an object.

---

**Route**

Edit, Links, select a link, Cancel Link.

---

---

**Example**

To delete the link to object PBRUSH.BMP, the command is:

```
OLECancelLink (LinkName: "PBRUSH.BMP")
```

---

## OLECancelLinks

**Purpose**

OLECancelLinks cancels a link updated or initialized between server and client applications.

**Syntax**

OLECancelLinks ()

**Route**

Edit, Links, Cancel.

---

## OLEChangeLink

**Purpose**

OLEChangeLink changes the file in a link.

**Syntax**

OLEChangeLink (ObjectName: *Character Expression*; Filename: *Character Expression*)

**Parameters**

*ObjectName*

Character Expression: Specifies an object name.

*Filename*

Character Expression: Specifies the name of the replacement file.

**Route**

Edit, Links, select a link, Change Link, select a new object.

**Example**

To specify a new linked file, the command is:

```
OLEChangeLink (ObjectName: "WPWIN60[0]"; Filename: "C:\WINDOWS\PBRUSH.BMP")
```

---

## OLEInitializeLinks

**Purpose**

OLEInitializeLinks reinitializes a link with its server application.

**IMPORTANT:** This command records when you create, update, change, or perform any function on a link.

---

|                 |                                                  |
|-----------------|--------------------------------------------------|
| <b>Syntax</b>   | OLEInitializeLinks ()                            |
| <b>Route</b>    | Edit, Links, select option.                      |
| <b>See Also</b> | ♦ OLEAcceptLinks ♦ OLEChangeLink ♦ OLEUpdateLink |

---

## OLEInsertDlg

---

|                |                                                     |
|----------------|-----------------------------------------------------|
| <b>Purpose</b> | OLEInsertDlg displays the Insert Object dialog box. |
| <b>Syntax</b>  | OLEInsertDlg ()                                     |
| <b>Route</b>   | Insert, Object.                                     |

---

## OLEInsertObject

---

|                   |                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OLEInsertObject specifies the name of a program to insert an object into.                                |
| <b>Syntax</b>     | OLEInsertObject (ObjectType: <i>Character Expression</i> )                                               |
| <b>Parameters</b> |                                                                                                          |
| <i>ObjectType</i> | Character Expression: Specifies the name of a program.                                                   |
| <b>Route</b>      | Insert, Object.                                                                                          |
| <b>Example</b>    | To specify the Paintbrush program, the command is:<br>OLEInsertObject (ObjectType: "Paintbrush Picture") |

---

## OLELinksDlg

---

|                |                                               |
|----------------|-----------------------------------------------|
| <b>Purpose</b> | OLELinksDlg displays the OLELinks dialog box. |
| <b>Syntax</b>  | OLELinksDlg ()                                |

---



---

|              |              |
|--------------|--------------|
| <b>Route</b> | Edit, Links. |
|--------------|--------------|

---

## OLEPasteDlg

---

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | OLEPasteDlg displays the Paste Special dialog box. |
|----------------|----------------------------------------------------|

---

|               |                |
|---------------|----------------|
| <b>Syntax</b> | OLEPasteDlg () |
|---------------|----------------|

---

|              |                      |
|--------------|----------------------|
| <b>Route</b> | Edit, Paste Special. |
|--------------|----------------------|

---

## OLESetLinkUpdate

---

|                |                                                           |
|----------------|-----------------------------------------------------------|
| <b>Purpose</b> | OLESetLinkUpdate specifies how to update a linked object. |
|----------------|-----------------------------------------------------------|

---

|               |                                                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | OLESetLinkUpdate (ObjectName: <i>Character Expression</i> ; UpdateType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|-------------------------------------------------------------------------------------------------------------------------------|

---

### Parameters

|                   |                                              |
|-------------------|----------------------------------------------|
| <i>ObjectName</i> | Character Expression: Specifies a link name. |
|-------------------|----------------------------------------------|

|                   |                                        |
|-------------------|----------------------------------------|
| <i>UpdateType</i> | Numeric Equivalent or Enumerated Type: |
|-------------------|----------------------------------------|

- 1 UpdateAlways!
- 2 UpdateManually!

---

|              |                                       |
|--------------|---------------------------------------|
| <b>Route</b> | Edit, Links, Automatic or Manual, OK. |
|--------------|---------------------------------------|

---

|                |                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To automatically update a link named Project, command is:<br>OLESetLinkUpdate (ObjectName: "Project"; UpdateType: UpdateAlways!) |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|

---

## OLEUpdateLink

---

|                |                                         |
|----------------|-----------------------------------------|
| <b>Purpose</b> | OLEUpdateLink updates a specified link. |
|----------------|-----------------------------------------|

---

|               |                                                        |
|---------------|--------------------------------------------------------|
| <b>Syntax</b> | OLEUpdateLink (LinkName: <i>Character Expression</i> ) |
|---------------|--------------------------------------------------------|

---

---

**Parameters**

*LinkName* Character Expression: Specifies the name of a link to update.

---

**Route** Edit, Links, Update Now.

---

**Example** To update the link WPWin60[0] the command is:

OLEUpdateLink (LinkName: "WPWin60[0]")

---

---

## OutlineBarSetup

**Purpose** OutlineBarSetup specifies whether to turn on the Outline bar.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineBarSetup (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display the Outline bar.

- 0 Off!
- 1 On!

---

**Example** To turn on the Outline bar, the command is:

OutlineBarSetup (State: On!)

---

---

## OutlineBeginNew

**Purpose** OutlineBeginNew begins a new outline in a specified style.

---

**Syntax** OutlineBeginNew (Style: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Style* Numeric Equivalent or Enumerated Type: Specifies a style for a new outline.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!

---

19 TableBoxNumberStyle!  
20 TextBoxNumberStyle!  
21 UserBoxNumberStyle!  
22 EquationNumberStyle!  
23 HypertextStyle!  
24 ToC1Style!  
25 ToC2Style!  
26 ToC3Style!  
27 ToC4Style!  
28 ToC5Style!  
29 Index1Style!  
30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 DocStle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!

- 67 Legal8Style!
- 68 Heading1Style!
- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

---

**Route** Tools, Outline, Options, Define Outline, Start New Outline.

---

**Example** To create a bullets style outline, the command is:  
OutlineBeginNew (Style: BulletsOutlnStyle!)

---

**See Also** ♦ OutlineEnd

---

## OutlineBodyText

---

**Purpose** OutlineBodyText hides or shows outline body text. When no parameter is specified, the command is a toggle.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineBodyText (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to show outline body text.

- 0 ShowBodyText!
- 1 HideBodyText!

---

**Example** To hide outline body text, the command is:  
OutlineBodyText (State: HideBodyText!)

---

**See Also** ♦ OutlineHideShowingBodyText

---

## OutlineBodyTextToggle

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineBodyTextToggle toggles selected text between an outline and body text. |
| <b>Syntax</b>   | OutlineBodyTextToggle ()                                                      |
| <b>Route</b>    | Tools, Outline, click the Body Text icon on the Outline feature bar.          |
| <b>See Also</b> | ♦ OutlineChangeToBodyText                                                     |

---

## OutlineChangeToBodyText

|                 |                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineChangeToBodyText changes selected text from an outline to body text.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command, use OutlineBodyTextToggle. |
| <b>Syntax</b>   | OutlineChangeToBodyText ()                                                                                                                                                                                                           |
| <b>See Also</b> | ♦ OutlineBodyText ♦ OutlineParagraphNumber                                                                                                                                                                                           |

---

## OutlineDefineDlg

|                |                                                          |
|----------------|----------------------------------------------------------|
| <b>Purpose</b> | OutlineDefineDlg displays the Outline Define dialog box. |
| <b>Syntax</b>  | OutlineDefineDlg ()                                      |
| <b>Route</b>   | Tools, Outline, Options button, Define Outline.          |

---

## OutlineDown

|                |                                                                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | OutlineDown blocks the next outline paragraph. After blocking each paragraph of an outline family, OutlineDown blocks the entire family.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Syntax** OutlineDown ()

---

**See Also** ♦ OutlineUp

---

---

## OutlineEnd

---

**Purpose** OutlineEnd turns off Outline.

---

**Syntax** OutlineEnd ()

---

**Route** Tools, Outline, Options, End Outline.

---

**See Also** ♦ OutlineBeginNew

---

---

## OutlineFamilyCopy

---

**Purpose** OutlineFamilyCopy copies an outline family and adjusts the numbering in the outline when the family is retrieved.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineFamilyCopy ()

---

**See Also** ♦ OutlineFamilyMove ♦ OutlineFamilyCut ♦ MoveModeEnd  
♦ OutlineFamilySelect

---

---

## OutlineFamilyCut

---

**Purpose** OutlineFamilyCut deletes an outline family and adjusts the numbering in the outline.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineFamilyCut ()

---

---

**See Also**

♦ OutlineFamilyMove ♦ OutlineFamilyCopy ♦ OutlineFamilySelect

---

## OutlineFamilyDemote

---

**Purpose**

OutlineFamilyDemote adjusts an outline family to the next level.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

OutlineFamilyDemote ()

---

## OutlineFamilyDown

---

**Purpose**

OutlineFamilyDown moves the insertion point to the end of the next outline family. In Edit Outline Mode, this command blocks the next outline family.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

OutlineFamilyDown ()

---

**See Also**

♦ OutlineFamilyUp

---

## OutlineFamilyDragDown

---

**Purpose**

OutlineFamilyDragDown positions a selected outline family below the next family and adjusts the numbering in Edit in Outline Mode.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable equivalent, use OutlineParagraphDragDown.

---

**Syntax**

OutlineFamilyDragDown ()

---

**Route**

Tools, Outline, Down Arrow button.

---

**See Also**

♦ OutlineFamilyUp ♦ OutlineFamilySelect ♦ OutlineParagraphDragDown

---

## OutlineFamilyDragUp

|                 |                                                                                                                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineFamilyDragUp moves a selected outline family above the previous family.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable equivalent, use OutlineParagraphDragUp. |
| <b>Syntax</b>   | OutlineFamilyDragUp ()                                                                                                                                                                                                                      |
| <b>Route</b>    | Tools, Outline, Up Arrow button.                                                                                                                                                                                                            |
| <b>See Also</b> | ♦ OutlineFamilyDragDown ♦ OutlineParagraphDragUp ♦ OutlineFamilySelect                                                                                                                                                                      |

---

## OutlineFamilyHide

|                   |                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineFamilyHide hides the current outline family.                                                                          |
| <b>Syntax</b>     | OutlineFamilyHide (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                              |
| <b>Parameters</b> |                                                                                                                              |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to display an outline family.<br>0 ShowBodyText!<br>1 HideBodyText! |
| <b>Route</b>      | Tools, Outline, Hide Family button (–).                                                                                      |
| <b>Example</b>    | To hide an outline family, the command is:<br>OutlineFamilyHide (State: HideBodyText!)                                       |
| <b>See Also</b>   | ♦ OutlineFamilyShow ♦ OutlineFamilySelect                                                                                    |

---

## OutlineFamilyLeft

|                |                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | OutlineFamilyLeft moves the insertion point from one outline paragraph to the previous outline paragraph.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



---

|                 |                      |
|-----------------|----------------------|
| <b>Syntax</b>   | OutlineFamilyLeft () |
| <b>See Also</b> | ♦ OutlineFamilyRight |

---

## OutlineFamilyMove

---

|                 |                                                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineFamilyMove moves an outline family.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineFamilyMove ()                                                                                                                               |
| <b>See Also</b> | ♦ OutlineFamilyCopy ♦ OutlineFamilyCut ♦ OutlineFamilySelect<br>♦ MoveModeEnd                                                                      |

---

## OutlineFamilyPromote

---

|                 |                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineFamilyPromote adjusts an outline family to the previous level in Edit in Outline Mode.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineFamilyPromote ()                                                                                                                                                                               |
| <b>See Also</b> | ♦ OutlineFamilyDemote ♦ OutlineFamilySelect                                                                                                                                                           |

---

## OutlineFamilyRight

---

|                |                                                                                                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | OutlineFamilyRight moves the insertion point from one outline paragraph to the next outline paragraph.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | OutlineFamilyRight ()                                                                                                                                                                                          |

---

---

**See Also** ♦ OutlineFamilyLeft

---

## OutlineFamilySelect

---

**Purpose** OutlineFamilySelect selects an outline family.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineFamilySelect ()

---

**See Also** ♦ OutlineParagraphSelect

---

## OutlineFamilyShow

---

**Purpose** OutlineFamilyShow displays a hidden outline family.

---

**Syntax** OutlineFamilyShow (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display a hidden outline family.

- 0 ShowBodyText!
- 1 HideBodyText!

---

**Route** Tools, Outline, Show Family button (+).

---

**Example** To show a hidden outline family, the command is:

OutlineFamilyShow (State: ShowBodyText!)

---

**See Also** ♦ OutlineFamilyHide

---

## OutlineFamilyUp

|                 |                                                                                                                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineFamilyUp moves the insertion point to the beginning of the previous outline family. With Select Mode on, this command blocks the previous outline family.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineFamilyUp ()                                                                                                                                                                                                                                                       |
| <b>See Also</b> | ♦ OutlineFamilyDown                                                                                                                                                                                                                                                      |

---

## OutlineHideShowingBodyText

|                 |                                                                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineHideShowingBodyText hides an outline but displays body text.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineHideShowingBodyText ()                                                                                                                                               |
| <b>See Also</b> | ♦ OutlineBodyText                                                                                                                                                           |

---

## OutlineNumberDlg

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | OutlineNumberDlg displays the Set Paragraph Number dialog box. |
| <b>Syntax</b>  | OutlineNumberDlg ()                                            |
| <b>Route</b>   | Tools, Outline, Options button, Set Number.                    |

---

## OutlineOff

|                |                                            |
|----------------|--------------------------------------------|
| <b>Purpose</b> | OutlineOff closes the Outline feature bar. |
| <b>Syntax</b>  | OutlineOff ()                              |

---

**Route** Tools, Outline, Close button.

---

## OutlineParagraphCopy

---

**Purpose** OutlineParagraphCopy copies an outline paragraph.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineParagraphCopy ()

---

**See Also** ♦ OutlineParagraphMove ♦ OutlineParagraphDelete ♦ MoveModeEnd  
♦ OutlineParagraphSelect

---

## OutlineParagraphDelete

---

**Purpose** OutlineParagraphDelete deletes an outline paragraph.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** OutlineParagraphDelete ()

---

**See Also** ♦ OutlineParagraphMove ♦ OutlineParagraphCopy ♦ MoveModeEnd  
♦ OutlineParagraphSelect

---

## OutlineParagraphDemote

---

**Purpose** OutlineParagraphDemote adjusts an outline paragraph to the next level.

---

**Syntax** OutlineParagraphDemote ()

---

**Route** Tools, Outline, Right Arrow button.

---

**See Also** ♦ OutlineParagraphPromote ♦ OutlineParagraphSelect

---

## OutlineParagraphDragDown

|                 |                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineParagraphDragDown positions an outline paragraph below the next paragraph and adjusts the numbering. |
| <b>Syntax</b>   | OutlineParagraphDragDown ()                                                                                 |
| <b>Route</b>    | Tools, Outline, Down Arrow button.                                                                          |
| <b>See Also</b> | ♦ OutlineParagraphDragUp ♦ OutlineParagraphSelect ♦ OutlineFamilyDragDown                                   |

---

## OutlineParagraphDragUp

|                 |                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineParagraphDragUp positions an outline paragraph above the previous paragraph and adjusts the numbering. |
| <b>Syntax</b>   | OutlineParagraphDragUp ()                                                                                     |
| <b>Route</b>    | Tools, Outline, Up Arrow button.                                                                              |
| <b>See Also</b> | ♦ OutlineParagraphDragDown ♦ OutlineParagraphSelect                                                           |

---

## OutlineParagraphMove

|                 |                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineParagraphMove moves an outline paragraph.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineParagraphMove ()                                                                                                                                  |
| <b>See Also</b> | ♦ OutlineParagraphCopy ♦ OutlineParagraphDelete ♦ MoveModeEnd<br>♦ OutlineParagraphSelect                                                                |

---

## OutlineParagraphNumber

|                   |                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineParagraphNumber specifies an outline paragraph level. If the Level parameter is not specified, the current number is inserted. |
| <b>Syntax</b>     | OutlineParagraphNumber (Level: <i>Numeric Expression</i> )                                                                            |
| <b>Parameters</b> |                                                                                                                                       |
| <i>Level</i>      | Numeric Expression: Specifies an outline paragraph level. Values range from 1 to 8.                                                   |
| <b>Route</b>      | Tools, Outline, Options button, Change Level, specify level.                                                                          |
| <b>Example</b>    | To adjust an outline paragraph to level 3, the command is:<br><code>OutlineParagraphNumber (Level: 3)</code>                          |

---

## OutlineParagraphPromote

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineParagraphPromote adjusts an outline paragraph to the previous level. |
| <b>Syntax</b>   | OutlineParagraphPromote ()                                                  |
| <b>Route</b>    | Tools, Outline, Right Arrow button.                                         |
| <b>See Also</b> | ♦ OutlineParagraphDemote ♦ OutlineParagraphSelect                           |

---

## OutlineParagraphSelect

|                 |                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineParagraphSelect selects an outline paragraph.<br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineParagraphSelect ()                                                                                                                                       |
| <b>See Also</b> | ♦ OutlineFamilySelect                                                                                                                                           |

---

## OutlineRight

|                 |                                                                                                                                                                                                                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | OutlineRight blocks the next outline family, with Select on in Outline mode. When repeated, the command unblocks the family and, with each repetition, blocks successive outline paragraphs until reaching the next family.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | OutlineRight ()                                                                                                                                                                                                                                                                                                                     |
| <b>See Also</b> | ♦ OutlineUp                                                                                                                                                                                                                                                                                                                         |

---

## OutlineSetParagraphNumber

|                   |                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineSetParagraphNumber sets the number for an outline paragraph.                                         |
| <b>Syntax</b>     | OutlineSetParagraphNumber (WordString: <i>Character Expression</i> )                                        |
| <b>Parameters</b> |                                                                                                             |
| <i>WordString</i> | Character Expression: Specifies an outline paragraph setting number.                                        |
| <b>Route</b>      | Tools, Outline, Options button, Set Number, enter a number or character.                                    |
| <b>Example</b>    | To set an outline paragraph number as 5, the command is:<br><br>OutlineSetParagraphNumber (WordString: "5") |

---

## OutlineShow

|                |                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | OutlineShow specifies the lowest outline level to show and whether to show body text.                        |
| <b>Syntax</b>  | OutlineShow (Level: <i>Numeric Expression</i> ; State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

## Parameters

|              |                                                                                                                                  |
|--------------|----------------------------------------------------------------------------------------------------------------------------------|
| <i>Level</i> | Numeric Expression: Specifies the lowest outline level to show. Values range from 1 to 8.                                        |
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display outline body text.<br><br>0 ShowBodyText!<br>1 HideBodyText! |

---

**Route** Tools, Outline, select Show button (1-8 or all).

---

**Example** To show outline levels 1 through 3 and hide the body text, the command is:  
  
OutlineShow (Level: 3; State: HideBodyText!)

---

**See Also** ♦ OutlineBodyText ♦ OutlineHideShowingBodyText

---

## OutlineStyleCopy

---

**Purpose** OutlineStyleCopy copies an outline style.

---

**Syntax** OutlineStyleCopy (Style: *Numeric Equivalent* or *Enumerated Type*, Source: *Numeric Equivalent* or *Enumerated Type*, Destination: *Numeric Equivalent* or *Enumerated Type*, NewName: *Character Expression*)

---

## Parameters

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i> | Numeric Equivalent or Enumerated Type: Specifies a style to copy.<br><br>0 NormalStyle!<br>16 FootnoteNumberInDocumentStyle!<br>17 EndnoteNumberInDocumentStyle!<br>18 FigureNumberStyle!<br>19 TableBoxNumberStyle!<br>20 TextBoxNumberStyle!<br>21 UserBoxNumberStyle!<br>22 EquationNumberStyle!<br>23 HypertextStyle!<br>24 ToC1Style!<br>25 ToC2Style!<br>26 ToC3Style!<br>27 ToC4Style!<br>28 ToC5Style!<br>29 Index1Style!<br>30 Index2Style! |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_20outlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

|                    |                                                                                                                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Source</i>      | Numeric Equivalent or Enumerated Type: Specifies the source of a specified style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                                                  |
| <i>Destination</i> | Numeric Equivalent or Enumerated Type: Specifies a destination to copy a style to.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                                                 |
| <i>NewName</i>     | Character Expression: Specifies a new style name.                                                                                                                                                             |
| <b>Route</b>       | Tools, Outline, Options button, Define Outline, select an outline, Options button, Copy, select option.                                                                                                       |
| <b>Example</b>     | To copy NormalStyle! from the current document to a personal library, the command is:<br><br>OutlineStyleCopy (Style: NormalStyle!; Source: CurrentDoc!; Destination: PersonalLibrary!; NewName: "New Style") |
| <b>See Also</b>    | ♦ OutlineStyleDelete ♦ OutlineStyleRetrieve ♦ OutlineStyleSave<br>♦ OutlineStyleSelect                                                                                                                        |

---

## OutlineStyleCreate

|                   |                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineStyleCreate creates, names, and saves an outline style.                                                                                                      |
| <b>Syntax</b>     | OutlineStyleCreate (Style: <i>Character Expression</i> ; Library: <i>Numeric Equivalent or Enumerated Type</i> )                                                    |
| <b>Parameters</b> |                                                                                                                                                                     |
| <i>Style</i>      | Character Expression: Specifies a new outline style.                                                                                                                |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies the library of a new outline style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                     |
| <b>Route</b>      | Tools, Outline, Options button, Define Outline, Create.                                                                                                             |
| <b>Example</b>    | To create an outline style named My Style and save it in the Shared Library, the command is:<br><br>OutlineStyleCreate (Style: "My Style"; Library: SharedLibrary!) |

---

## OutlineStyleDelete

---

**Purpose**

OutlineStyleDelete deletes an outline style.

**Syntax**

OutlineStyleDelete (Style: *Numeric Equivalent* or *Enumerated Type*; Library: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*Style* Numeric Equivalent or Enumerated Type: Specifies a style to delete.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutlnStyle!
- 46 OutlineStyle!
- 47 LegalOutlnStyle!

48 BulletsOutlnStyle!  
 49 HeadingsOutlnStyle!  
 50 Legal\_20outlnStyle!  
 51 NumbersOutlnStyle!  
 52 Level1Style!  
 53 Level2Style!  
 54 Level3Style!  
 55 Level4Style!  
 56 Level5Style!  
 57 Level6Style!  
 58 Level7Style!  
 59 Level8Style!  
 60 Legal1Style!  
 61 Legal2Style!  
 62 Legal3Style!  
 63 Legal4Style!  
 64 Legal5Style!  
 65 Legal6Style!  
 66 Legal7Style!  
 67 Legal8Style!  
 68 Heading1Style!  
 69 Heading2Style!  
 70 Heading3Style!  
 71 Heading4Style!  
 72 Heading5Style!  
 73 Heading6Style!  
 74 Heading7Style!  
 75 Heading8Style!

*Library*

Numeric Equivalent or Enumerated Type: Specifies where to delete an outline style.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route**

Tools, Outline, Options, select an Outline Style, Options, Delete.

---

**Example**

To delete Newstyle from the current document, the command is:

OutlineStyleDelete (Style: "Newstyle"; Library: CurrentDoc!)

---

**See Also**

• OutlineStyleCopy • OutlineStyleRetrieve • OutlineStyleSave • OutlineStyleSelect

---

## OutlineStyleDescription

---

|                    |                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | OutlineStyleDescription describes an outline style.                                                                                            |
| <b>Syntax</b>      | OutlineStyleDescription (Description: <i>Character Expression</i> )                                                                            |
| <b>Parameters</b>  |                                                                                                                                                |
| <i>Description</i> | Character Expression: Specifies a description for an outline style.                                                                            |
| <b>Route</b>       | Tools, Outline, Options, Define Outline, select an outline, Edit, Description, type description.                                               |
| <b>Example</b>     | To describe an outline style as Table of Contents Format, the command is:<br>OutlineStyleDescription (Description: "Table of Contents Format") |
| <b>See Also</b>    | ♦ OutlineStyleEditBegin ♦ OutlineStyleEditEnd                                                                                                  |

---

## OutlineStyleEditBegin

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineStyleEditBegin is the opening command of an OutlineStyleEditBegin-OutlineStyleEditEnd statement.                                                                                                                                                                                                                                                                                                                                |
| <b>Syntax</b>     | OutlineStyleEditBegin (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>Style</i>      | Numeric Equivalent or Enumerated Type: Specifies an outline style to edit.<br>0 NormalStyle!<br>16 FootnoteNumberInDocumentStyle!<br>17 EndnoteNumberInDocumentStyle!<br>18 FigureNumberStyle!<br>19 TableBoxNumberStyle!<br>20 TextBoxNumberStyle!<br>21 UserBoxNumberStyle!<br>22 EquationNumberStyle!<br>23 HypertextStyle!<br>24 ToC1Style!<br>25 ToC2Style!<br>26 ToC3Style!<br>27 ToC4Style!<br>28 ToC5Style!<br>29 Index1Style! |

30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 InitialCodesStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

|                 |                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Library</i>  | Numeric Equivalent or Enumerated Type: Specifies the library of an outline style.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!               |
| <b>Route</b>    | Tools, Outline, Options, Define Outline, select a style, Edit.                                                                                             |
| <b>Example</b>  | To edit an outline style named Bullets in the current document, the command is:<br>OutlineStyleEditBegin (Style: BulletsOutInStyle!; Library: CurrentDoc!) |
| <b>See Also</b> | ♦ OutlineStyleEditEnd                                                                                                                                      |

---

## OutlineStyleEditEnd

|                   |                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineStyleEditEnd is the closing command of an OutlineStyleEditBegin-OutlineStyleEditEnd statement.           |
| <b>Syntax</b>     | OutlineStyleEditEnd (State: <i>Numeric Equivalent or Enumerated Type</i> )                                      |
| <b>Parameters</b> |                                                                                                                 |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to save outline style changes.<br>0 Cancel!<br>1 Save! |
| <b>Route</b>      | Tools, Outline, Options button, Define Outline, select a style, Edit.                                           |
| <b>Example</b>    | To save outline style changes, the command is:<br>OutlineStyleEditEnd (State: Save!)                            |
| <b>See Also</b>   | ♦ OutlineStyleEditBegin                                                                                         |

---

# OutlineStyleLevel

---

**Purpose** OutlineStyleLevel specifies an outline level and a style for that level.

---

**Syntax** OutlineStyleLevel (Level: *Numeric Expression*; Style: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*Level* Numeric Expression: Specifies an outline level. Values range from 1 to 8.

*Style* Numeric Equivalent or Enumerated Type: Specifies an outline style for a selected level.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 InitialCodesStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutlnStyle!
- 46 OutlineStyle!
- 47 LegalOutlnStyle!
- 48 BulletsOutlnStyle!
- 49 HeadingsOutlnStyle!



50 Legal\_2OutlineStyle!  
51 NumbersOutlineStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

---

**Route** Tools, Outline, Options button, Define Outline, select a style, Edit, select a level, select an associated style.

---

**Example** To apply a style named Heading 1 to outline level 3, the command is:  
`OutlineStyleLevel (Level: 3; Style: Heading1Style!)`

---

## OutlineStyleNumberFormat

---

**Purpose** OutlineStyleNumberFormat specifies the numbering format for each level of a selected outline style.

---

**Syntax** OutlineStyleNumberFormat (Level: *Numeric Expression*; Format: *Character Expression*)

---

**Parameters**

|               |                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Level</i>  | Numeric Expression: Specifies an outline level. Values range from 1 to 8.                                                                                  |
| <i>Format</i> | Character Expression: Specifies the numbering format for each outline level. Select a system format or enter a character expression specifying the format. |

---

|              |                                                                                                                                                            |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Tools, Outline, Options button, Define Outline, select a style, Edit, select a level, Custom Number, enter or choose a number format from the pop-up list. |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                |                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify paragraph numbering style for level 1, the command is:<br><code>OutlineStyleNumberFormat (Level: 1; Format: "1.")</code> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                        |
|-----------------|----------------------------------------|
| <b>See Also</b> | ♦ <code>OutlineStyleNumbersOnly</code> |
|-----------------|----------------------------------------|

---

## OutlineStyleNumbersOnly

---

|                |                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>OutlineStyleNumbersOnly</code> specifies whether to apply the Numbers Only format to a selected outline. |
|----------------|----------------------------------------------------------------------------------------------------------------|

---

|               |                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | <code>OutlineStyleNumbersOnly</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|----------------------------------------------------------------------------------------------------|

---

**Parameters**

|              |                                                                                                                                                                                                   |   |     |   |      |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----|---|------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether to define level styles or use the Numbers Only format.<br><table><tr><td>0</td><td>No!</td></tr><tr><td>1</td><td>Yes!</td></tr></table> | 0 | No! | 1 | Yes! |
| 0            | No!                                                                                                                                                                                               |   |     |   |      |
| 1            | Yes!                                                                                                                                                                                              |   |     |   |      |

---

|              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| <b>Route</b> | Tools, Outline, Options button, Define Outline, select a style, Edit, Options, Numbers Only. |
|--------------|----------------------------------------------------------------------------------------------|

---

|                |                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify the Numbers Only format, the command is:<br><code>OutlineStyleNumbersOnly (State: Yes!)</code> |
|----------------|-----------------------------------------------------------------------------------------------------------|

---

|                 |                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| <b>See Also</b> | ♦ <code>OutlineStyleNumberFormat</code> ♦ <code>OutlineStyleEditBegin</code> ♦ <code>OutlineStyleEditEnd</code> |
|-----------------|-----------------------------------------------------------------------------------------------------------------|

---

# OutlineStyleRename

---

|                   |                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineStyleRename renames a user-defined outline style.                                                                                 |
| <b>Syntax</b>     | OutlineStyleRename (Name: <i>Character Expression</i> )                                                                                  |
| <b>Parameters</b> |                                                                                                                                          |
| <i>Name</i>       | Character Expression: Specifies a new name for a user-defined outline style.                                                             |
| <b>Route</b>      | Tools, Outline, Options, Define Outline, select a user-defined style, Edit, Name, type a new name.                                       |
| <b>Example</b>    | After selecting a user-defined outline style, to rename the style My Style, the command is:<br><br>OutlineStyleRename (Name: "My Style") |
| <b>See Also</b>   | ♦ OutlineStyleEditBegin ♦ OutlineStyleEditEnd                                                                                            |

---

# OutlineStyleRetrieve

---

|                   |                                                                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineStyleRetrieve retrieves outline styles from the current document or a specified file.                                                                                                            |
| <b>Syntax</b>     | OutlineStyleRetrieve (Filename: <i>Character Expression</i> ; Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                |
| <b>Parameters</b> |                                                                                                                                                                                                         |
| <i>Filename</i>   | Character Expression: Specifies the name of a file to retrieve outline styles from.                                                                                                                     |
| <i>Type</i>       | Numeric Equivalent or Enumerated Type: Specifies which outline styles to retrieve.<br><ol style="list-style-type: none"><li>1 UserStyles!</li><li>2 SystemStyles!</li><li>3 AllStyles!</li></ol>        |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies where to retrieve outline styles.<br><ol style="list-style-type: none"><li>0 CurrentDoc!</li><li>1 PersonalLibrary!</li><li>2 SharedLibrary!</li></ol> |

|                 |                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Tools, Outline, Options button, select an outline style, Options button, Retrieve, type a filename.                                                                                   |
| <b>Example</b>  | To retrieve outline styles from the file MYSTYLE into the current document, the command is:<br><br>OutlineStyleRetrieve (Filename: "MYSTYLE"; Type: AllStyles!; Library: CurrentDoc!) |
| <b>See Also</b> | ♦ OutlineStyleCopy ♦ OutlineStyleDelete ♦ OutlineStyleSave ♦ OutlineStyleSelect                                                                                                       |

---

## OutlineStyleSave

|                   |                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | OutlineStyleSave saves an outline style.                                                                                                                                                                              |
| <b>Syntax</b>     | OutlineStyleSave (Filename: <i>Character Expression</i> ; Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                  |
| <b>Parameters</b> |                                                                                                                                                                                                                       |
| <i>Filename</i>   | Character Expression: Specifies the name of a file to save outline styles to.                                                                                                                                         |
| <i>Type</i>       | Numeric Equivalent or Enumerated Type: Specifies which styles to save.<br><ol style="list-style-type: none"> <li>1 UserStyles!</li> <li>2 SystemStyles!</li> <li>3 AllStyles!</li> </ol>                              |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies the location of the outline styles to save.<br><ol style="list-style-type: none"> <li>0 CurrentDoc!</li> <li>1 PersonalLibrary!</li> <li>2 SharedLibrary!</li> </ol> |
| <b>Route</b>      | Tools, Outline, Options, select an outline style, Options, Save As, enter a filename.                                                                                                                                 |
| <b>Example</b>    | To save all outline styles in the current document to the file NEWSTYLE.WPT, the command is:<br><br>OutlineStyleSave (Filename: "NEWSTYLE.WPT"; Type: AllStyles!; Library: CurrentDoc!)                               |
| <b>See Also</b>   | ♦ OutlineStyleCopy ♦ OutlineStyleDelete ♦ OutlineStyleRetrieve<br>♦ OutlineStyleSelect                                                                                                                                |

---

# OutlineStyleSelect

---

**Purpose** OutlineStyleSelect selects a specified outline style.

---

**Syntax** OutlineStyleSelect (Style: *Numeric Equivalent or Enumerated Type or Character Expression*)

---

## Parameters

*Style* Numeric Equivalent or Enumerated Type or Character Expression: Specifies a style. Use a character expression to specify a user-defined style.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TextBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutlnStyle!
- 46 OutlineStyle!
- 47 LegalOutlnStyle!
- 48 BulletsOutlnStyle!
- 49 HeadingsOutlnStyle!
- 50 Legal\_20utlnStyle!

- 51 NumbersOutLnStyle!
- 52 Level1Style!
- 53 Level2Style!
- 54 Level3Style!
- 55 Level4Style!
- 56 Level5Style!
- 57 Level6Style!
- 58 Level7Style!
- 59 Level8Style!
- 60 Legal1Style!
- 61 Legal2Style!
- 62 Legal3Style!
- 63 Legal4Style!
- 64 Legal5Style!
- 65 Legal6Style!
- 66 Legal7Style!
- 67 Legal8Style!
- 68 Heading1Style!
- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

---

**Route** Tools, Outline, Options button, Define Outline, select style.

---

**Example** To select HeadingsOutLnStyle!, the command is:

OutlineStyleSelect (Style: HeadingsOutLnStyle!)

---

**See Also** ♦ OutlineStyleCopy ♦ OutlineStyleDelete ♦ OutlineStyleRetrieve  
♦ OutlineStyleSave

---

## OutlineUp

---

**Purpose** OutlineUp blocks the previous outline paragraph. After blocking each paragraph of an outline family, OutlineUp blocks the entire family.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

|                 |               |
|-----------------|---------------|
| <b>Syntax</b>   | OutlineUp ()  |
| <b>See Also</b> | ♦ OutlineDown |

---

## OutputOption

---

**Purpose** OutputOption specifies a printer output type.

**Syntax** OutputOption (OutputType: *Numeric Equivalent* or *Enumerated Type*, BinsToMark: *Numeric Expression*)

### Parameters

*OutputType* Numeric Equivalent or Enumerated Type: Specifies the type of output from a printer.

- 1 OutputToSort!
- 2 OutputToPutGroup!
- 16 OutputToBins!
- 64 OutputPutNone!

*BinsToMark* Numeric Expression: Specifies output bins to mark.

**Route** File, Print, Options, select options.

**Example** To sort output into bins 1 and 2, the command is:  
OutputOption (OutputType: OutputToSort!; BinsToMark: 1, 2)

**See Also** ♦ OutputOptionsSetOffsetJogger

---

## OutputOptionsSetOffsetJogger

---

**Purpose** OutputOptionsSetOffsetJogger specifies whether Jogger is selected in the Output Bin Options dialog box.

**Syntax** OutputOptionsSetOffsetJogger (State: *Numeric Equivalent* or *Enumerated Type*)

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether Jogger is active.

- 8 JoggerOn!
- 10 JoggerOff!

---

|                |                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | File, Print, Options, Output Bin Options, select or deselect Jogger.                                          |
| <b>Example</b> | To turn on the Offset Jogger, the command is:<br><code>OutputOptionsSetOffsetJogger (State: JoggerOn!)</code> |

---

## OverstrikeCreate

---

|                    |                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | OverstrikeCreate creates an overstrike.                                                                 |
| <b>Syntax</b>      | OverstrikeCreate (OvrStrkStrg: <i>Character Expression</i> )                                            |
| <b>Parameters</b>  |                                                                                                         |
| <i>OvrStrkStrg</i> | Character Expression: Specifies two or more characters to create an overstrike character.               |
| <b>Route</b>       | Layout, Typesetting, Overstrike, enter overstrike characters.                                           |
| <b>Example</b>     | To overstrike characters = and \, the command is:<br><code>OverstrikeCreate (OvrStrkStrg: "=\\")</code> |
| <b>See Also</b>    | ♦ OverstrikeCreateDlg ♦ OverstrikeEdit ♦ ?Overstrike                                                    |

---

## OverstrikeCreateDlg

---

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| <b>Purpose</b>  | OverstrikeCreateDlg displays the Overstrike dialog box. |
| <b>Syntax</b>   | OverstrikeCreateDlg ()                                  |
| <b>Route</b>    | Layout, Typesetting, Overstrike.                        |
| <b>See Also</b> | ♦ OverstrikeCreate ♦ OverstrikeEdit                     |

---



---

## OverstrikeEdit

|                    |                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | OverstrikeEdit edits the first overstrike to the right of the insertion point. The edited overstrike is replaced by the text in the OvrStrkStrg parameter. |
| <b>Syntax</b>      | OverstrikeEdit (OvrStrkStrg: <i>Character Expression</i> )                                                                                                 |
| <b>Parameters</b>  |                                                                                                                                                            |
| <i>OvrStrkStrg</i> | Character Expression: Specifies new overstrike characters.                                                                                                 |
| <b>Route</b>       | Layout, Typesetting, Overstrike, Next, edit overstrike characters.                                                                                         |
| <b>Example</b>     | To replace an existing overstrike with characters x, y, and z, the command is:<br>OverstrikeEdit (OvrStrkStrg: "xyz")                                      |
| <b>See Also</b>    | ♦ OverstrikeCreate ♦ OverstrikeCreateDlg                                                                                                                   |

---

## PageBorderDlg

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <b>Purpose</b>  | PageBorderDlg displays the Page Border dialog box. |
| <b>Syntax</b>   | PageBorderDlg ()                                   |
| <b>Route</b>    | Layout, Page, Border/Fill.                         |
| <b>See Also</b> | ♦ ?PageBorder                                      |

---

## PageNumber

|                   |                                                                  |
|-------------------|------------------------------------------------------------------|
| <b>Purpose</b>    | PageNumber specifies a new page number for the current page.     |
| <b>Syntax</b>     | PageNumber (Page: <i>Numeric Expression</i> )                    |
| <b>Parameters</b> |                                                                  |
| <i>Page</i>       | Numeric Expression: Specifies a new number for the current page. |

---

|                |                                                                                           |
|----------------|-------------------------------------------------------------------------------------------|
| <b>Route</b>   | Layout, Page, Numbering, Value, specify a new page number.                                |
| <b>Example</b> | To change the number of a page to 5, the command is:<br><code>PageNumber (Page: 5)</code> |

---

## PageNumberDecrement

---

|                 |                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PageNumberDecrement decreases the current page number by one.                                                      |
| <b>Syntax</b>   | PageNumberDecrement ()                                                                                             |
| <b>Route</b>    | Layout, Page, Numbering, Value, specify a negative number in the Increase/Decrease Existing Page Number entry box. |
| <b>See Also</b> | ♦ PageNumberIncrement                                                                                              |

---

## PageNumberDisplay

---

|                |                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PageNumberDisplay displays a page number in the current document window at the insertion point.              |
| <b>Syntax</b>  | PageNumberDisplay ()                                                                                         |
| <b>Route</b>   | Layout, Page, Numbering, Value, select Insert and Display at Insertion Point in the Page Settings group box. |

---

## PageNumberDisplayFormat

---

|                |                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PageNumberDisplayFormat displays a formatted page number in the current document window.         |
| <b>Syntax</b>  | PageNumberDisplayFormat ()                                                                       |
| <b>Route</b>   | Layout, Page, Numbering, Options, select Insert Format and Accompanying Text at Insertion Point. |

---

---

**See Also** ♦ PageNumberFormat

---

## PageNumberFormat

**Purpose** PageNumberFormat specifies a page number format including text, volume or chapter numbers, or secondary page numbers.

**Syntax** PageNumberFormat (Format: *Character Expression*)

**Parameters**

*Format* Character Expression: Specifies a page number format.

**Route** Layout, Page, Numbering, Options, select an option from the Insert pop-up list.

**Example** To format page numbers as Page n (n being the number), the command is:  
PageNumberFormat (Format: "Page [page #]")

**See Also** ♦ PageNumberDisplayFormat

---

## PageNumberIncrement

**Purpose** PageNumberIncrement increases the current page number by one.

**Syntax** PageNumberIncrement ()

**Route** Layout, Page, Numbering, Value, specify a positive number in the Increase/Decrease Existing Page Number entry box.

**See Also** ♦ PageNumberDecrement

---

## PageNumberingDlg

**Purpose** PageNumberingDlg displays the Page Numbering dialog box.

**Syntax** PageNumberingDlg ()

---

**Route** Layout, Page, Numbering.

---

## PageNumberMethod

---

**Purpose** PageNumberMethod specifies a page numbering method.

---

**Syntax** PageNumberMethod (Method: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Method* Numeric Equivalent or Enumerated Type: Specifies a page numbering method.

- 0 Numbers!
- 1 LowerLetters!
- 2 UpperLetters!
- 3 LowerRoman!
- 4 UpperRoman!

---

**Route** Layout, Page, Numbering, Options, select an option from the Page: Numbers pop-up list.

---

**Example** To specify uppercase roman numerals as the page numbering method, the command is:

PageNumberMethod (Method: UpperRoman!)

---

**See Also** ♦ ?PageNumberMethod ♦ ?ParagraphNumberMethod

---

## PageNumberPosition

---

**Purpose** PageNumberPosition specifies the position and appearance of page numbering.

---

**Syntax** PageNumberPosition (Position: *Numeric Equivalent* or *Enumerated Type*; Default: *Numeric Equivalent* or *Enumerated Type*; Font: *Character Expression*; FontSize: *Numeric Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; Shade: *Numeric Expression*; {Attributes: *Numeric Equivalent* or *Enumerated Type*})

---

**Parameters**

*Position* Numeric Equivalent or Enumerated Type: Specifies where to print a page number.

- 0 None!
- 1 TopLeft!
- 2 TopCenter!

- 3 TopRight!
- 4 TopAlternating!
- 5 BottomLeft!
- 6 BottomCenter!
- 7 BottomRight!
- 8 BottomAlternating!

*Default* Numeric Equivalent or Enumerated Type: Specifies whether the page numbering font uses the initial document font.

- 0 UseDefaultValues!
- 1 DontUseDefaultValues!

*Font* Character Expression: Specifies a page numbering font. Use this parameter only if the Default parameter is set to DontUseDefaultValues! or 1.

*FontSize* Measurement Expression: Specifies the point size of the page numbering font. Use this parameter if the Default parameter is set to DontUseDefaultValues! or 1.

*Red* Numeric Expression: Specifies the amount of red in a color. Values range from 0 to 255. Use this parameter only if the Default parameter is set to DontUseDefaultValues! or 1.

*Green* Numeric Expression: Specifies the amount of green in a color. Values range from 0 to 255. Use this parameter only if the Default parameter is set to DontUseDefaultValues! or 1.

*Blue* Numeric Expression: Specifies the amount of blue in a color. Values range from 0 to 255. Use this parameter only if the Default parameter is set to DontUseDefaultValues! or 1.

*Shade* Numeric Expression: Specifies the shading percentage of a color. Values range from 0 to 100. Use this parameter only if the Default parameter is set to DontUseDefaultValues! or 1.

*Attributes* Numeric Equivalent or Enumerated Type: Specifies font attributes and relative size. This parameter is placed in braces {}. You may specify more than one attribute.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 Superscript!
- 6 Subscript!
- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!

15 SmallCaps!

16 Every!

---

**Route** Layout, Page, Numbering, select an option from the Position pop-up list, Font, select options.

---

**Example** To position page numbers at the bottom center of each page in a small, italicized, bold font, the command is:

```
PageNumberPosition (Position: BottomCenter!; Default: DontUseDefaultValues!; Font: "Courier";
FontSize: 12; Red: 0; Green: 0; Blue: 255; Shade: 0; {Attributes: Small!; Attributes: Italics!; Attributes:
Bold!})
```

---

**See Also** ♦ ?PageNumberPosition

---

## PageNumberSettingsDlg

---

**Purpose** PageNumberSettingsDlg displays the Numbering Value dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PageNumberSettingsDlg ()

---

## PageSuppressDlg

---

**Purpose** PageSuppressDlg displays the Suppress dialog box.

---

**Syntax** PageSuppressDlg ()

---

**Route** Layout, Page, Suppress.

---

**See Also** ♦ ?PageSuppress

---

## PageZoomFull

|                 |                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PageZoomFull displays a full view of the current page.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | PageZoomFull ()                                                                                                                                            |
| <b>See Also</b> | ♦ DisplayZoom                                                                                                                                              |

---

## PaneNext

|                |                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PaneNext moves focus to the next pane in the equation editor.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | PaneNext ()                                                                                                                                                       |

---

## PanePrevious

|                |                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PanePrevious moves focus to the previous pane in the equation editor.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | PanePrevious ()                                                                                                                                                           |

---

## PaperSizeAdd

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PaperSizeAdd creates a Paper Size/Type definition for the Paper Size dialog box.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Syntax</b>  | PaperSizeAdd (Name: <i>Character Expression</i> ; Width: <i>Measurement Expression</i> ; Length: <i>Measurement Expression</i> ; WidthAdjust: <i>Measurement Expression</i> ; LengthAdjust: <i>Measurement Expression</i> ; Flags: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Location: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Use: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; ReturnAddressHoriz: <i>Numeric</i> ) |

*Expression; ReturnAddressVert: Numeric Expression; MailingAddressHoriz: Numeric Expression; MailingAddressVert: Numeric Expression)*

---

## Parameters

|                     |                                                                                                                                                                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Name</i>         | Character Expression: Specifies a paper type name if the Type parameter is Other!.                                                                                                                                                              |
| <i>Width</i>        | Measurement Expression: Specifies page width.                                                                                                                                                                                                   |
| <i>Length</i>       | Measurement Expression: Specifies page length.                                                                                                                                                                                                  |
| <i>WidthAdjust</i>  | Measurement Expression: Adjusts page contents left or right with a positive or negative value to this parameter.                                                                                                                                |
| <i>LengthAdjust</i> | Measurement Expression: Adjust page contents to up or down with a positive or negative value to this parameter.                                                                                                                                 |
| <i>Flags</i>        | Numeric Equivalent or Enumerated Type: Specifies orientation.<br>1 Portrait!<br>2 Landscape!<br>4 PrimaryForm!<br>8 FormCurrentlyLoaded!<br>9 PortraitLoaded!<br>10 LandscapeLoaded!<br>13 PortraitPrimaryLoaded!<br>14 LandscapePrimaryLoaded! |
| <i>Location</i>     | Numeric Equivalent or Enumerated Type: Specifies how paper is fed to a printer.<br>0 Continuous!<br>32 Manual!                                                                                                                                  |
| <i>Type</i>         | Numeric Equivalent or Enumerated Type: Specifies a paper type.<br>0 AllOthers!<br>1 Standard!<br>2 Bond!<br>3 Letterhead!<br>4 Labels!<br>5 Envelope!<br>6 Transparency!<br>7 Cardstock!<br>8 Glossy!<br>9 ClayBased!<br>255 Other!             |
| <i>Use</i>          | Numeric Equivalent or Enumerated Type:<br>0 None!<br>1 US!<br>2 Europe!<br>4 Latin_America!<br>8 Japan!                                                                                                                                         |



|                            |                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ReturnAddressHoriz</i>  | Numeric Expression: Specifies the return address horizontal position if the Type parameter is Envelope!.                                                                                                                                                                                                                                                 |
| <i>ReturnAddressVert</i>   | Numeric Expression: Specifies the return address vertical position if the Type parameter is Envelope!.                                                                                                                                                                                                                                                   |
| <i>MailingAddressHoriz</i> | Numeric Expression: Specifies the mailing address horizontal position if the Type parameter is Envelope!.                                                                                                                                                                                                                                                |
| <i>MailingAddressVert</i>  | Numeric Expression: Specifies the mailing address vertical position if the Type parameter is Envelope!.                                                                                                                                                                                                                                                  |
| <b>Route</b>               | Layout, Page, Paper Size, Create, type a Paper Name and specify Size, Paper Location, Orientation, and Text Adjustment options.                                                                                                                                                                                                                          |
| <b>Example</b>             | To add a paper size/type named New Size, the command is:<br><br><pre>PaperSizeAdd (Name: "New Size"; Width: 8.5"; Length: 11"; WidthAdjust: .25"; LengthAdjust: .2"; Flags: Portrait!; Location: Continuous!; Type: Standard!; Use: None!; ReturnAddressHoriz: 0.0"; ReturnAddressVert: 0.0"; MailingAddressHoriz: 0.0"; MailingAddressVert: 0.0")</pre> |
| <b>See Also</b>            | ♦ PaperSizeDelete ♦ PaperSizeResave ♦ ?PaperSizeType                                                                                                                                                                                                                                                                                                     |

---

## PaperSizeDelete

|                   |                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PaperSizeDelete deletes a specified Paper Size/Type definition.                                         |
| <b>Syntax</b>     | PaperSizeDelete (Name: <i>Character Expression</i> )                                                    |
| <b>Parameters</b> |                                                                                                         |
| <i>Name</i>       | Character Expression: Specifies a paper size to delete.                                                 |
| <b>Route</b>      | Layout, Page, Paper Size, select a paper definition, Delete.                                            |
| <b>Example</b>    | To delete a paper size named Newsiz, the command is:<br><br><pre>PaperSizeDelete (Name: "Newsiz")</pre> |
| <b>See Also</b>   | ♦ PaperSizeAdd ♦ PaperSizeDlg                                                                           |

---

# PaperSizeDlg

---

|                 |                                                   |
|-----------------|---------------------------------------------------|
| <b>Purpose</b>  | PaperSizeDlg displays the Paper Size dialog box.  |
| <b>Syntax</b>   | PaperSizeDlg ()                                   |
| <b>Route</b>    | Layout, Page, Paper Size.                         |
| <b>See Also</b> | ♦ PaperSizeAdd ♦ PaperSizeDelete ♦ ?PaperSizeType |

---

# PaperSizeResave

---

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PaperSizeResave edits and saves a Paper Size/Type definition for the Paper Size dialog box.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Syntax</b>  | PaperSizeResave (OldName: <i>Character Expression</i> ; NewName: <i>Character Expression</i> ; Width: <i>Measurement Expression</i> ; Length: <i>Measurement Expression</i> ; WidthAdjust: <i>Measurement Expression</i> ; LengthAdjust: <i>Measurement Expression</i> ; Flags: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Location: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Use: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; ReturnAddressHoriz: <i>Numeric Expression</i> ; ReturnAddressVert: <i>Numeric Expression</i> ; MailingAddressHoriz: <i>Numeric Expression</i> ; MailingAddressVert: <i>Numeric Expression</i> ) |

---

## Parameters

|                     |                                                                                                                                                                                                                                                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>OldName</i>      | Character Expression: Specifies the old paper type name.                                                                                                                                                                                                      |
| <i>NewName</i>      | Character Expression: Specifies the new paper type name.                                                                                                                                                                                                      |
| <i>Width</i>        | Measurement Expression: Specifies page width.                                                                                                                                                                                                                 |
| <i>Length</i>       | Measurement Expression: Specifies page length.                                                                                                                                                                                                                |
| <i>WidthAdjust</i>  | Measurement Expression: Adjusts page contents left or right with a positive or negative value to this parameter.                                                                                                                                              |
| <i>LengthAdjust</i> | Measurement Expression: Adjust page contents to up or down with a positive or negative value to this parameter.                                                                                                                                               |
| <i>Flags</i>        | Numeric Equivalent or Enumerated Type: Specifies orientation.<br><ul style="list-style-type: none"><li>1 Portrait!</li><li>2 Landscape!</li><li>4 PrimaryForm!</li><li>8 FormCurrentlyLoaded!</li><li>9 PortraitLoaded!</li><li>10 LandscapeLoaded!</li></ul> |

|                            |                                                                                                                                                                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | 13 PortraitPrimaryLoaded!                                                                                                                                                                                                           |
|                            | 14 LandscapePrimaryLoaded!                                                                                                                                                                                                          |
| <i>Location</i>            | Numeric Equivalent or Enumerated Type: Specifies how paper is fed to a printer.<br>0 Continuous!<br>32 Manual!                                                                                                                      |
| <i>Type</i>                | Numeric Equivalent or Enumerated Type: Specifies a paper type.<br>0 AllOthers!<br>1 Standard!<br>2 Bond!<br>3 Letterhead!<br>4 Labels!<br>5 Envelope!<br>6 Transparency!<br>7 Cardstock!<br>8 Glossy!<br>9 ClayBased!<br>255 Other! |
| <i>Use</i>                 | Numeric Equivalent or Enumerated Type<br>0 None!<br>1 US!<br>2 Europe!<br>4 Latin_America!<br>8 Japan!                                                                                                                              |
| <i>ReturnAddressHoriz</i>  | Numeric Expression: Specifies the return address horizontal position if the Type parameter is Envelope!.                                                                                                                            |
| <i>ReturnAddressVert</i>   | Numeric Expression: Specifies the return address vertical position if the Type parameter is Envelope!.                                                                                                                              |
| <i>MailingAddressHoriz</i> | Numeric Expression: Specifies the mailing address horizontal position if the Type parameter is Envelope!.                                                                                                                           |
| <i>MailingAddressVert</i>  | Numeric Expression: Specifies the mailing address vertical position if the Type parameter is Envelope!.                                                                                                                             |

---

**Route** Layout, Page, PaperSize, select a paper definition, Edit, specify changes.

---

**Example** To resave a paper size, the command is:

```
PaperSizeResave (OldName: "Letter"; NewName: "Letter Special"; Width: 8.5"; Length: 11"; WidthAdjust: 0.0"; LengthAdjust: 0.0"; Flags: PortraitPrimaryLoaded!; Location: Continuous!; Type: Letterhead!; Use: None!ReturnAddressHoriz: 0.0"; ReturnAddressVert: 0.0"; MailingAddressHoriz: 0.0"; MailingAddressVert: 0.0")
```

---

**See Also** ♦ PaperSizeAdd ♦ PaperSizeDelete

---

## PaperSizeSelect

---

**Purpose** PaperSizeSelect specifies a paper size for the current document.

---

**Syntax** PaperSizeSelect (Name: *Character Expression*; Width: *Numeric Expression*; Height: *Numeric Expression*; Type: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Name* Character Expression: Specifies the name of a paper definition.

*Width* Numeric Expression: Specifies a paper width.

*Height* Numeric Expression: Specifies a paper height.

*Type* Numeric Equivalent or Enumerated Type: Specifies a paper type.

- 1 Standard!
- 2 Bond!
- 3 Letterhead!
- 4 Labels!
- 5 Envelope!
- 6 Transparency!
- 7 Cardstock!
- 8 Glossy!
- 9 ClayBased!
- 255 Other!

---

**Route** Layout, Page, Paper Size, highlight a paper definition, Select.

---

**Example** To select B5 Landscape paper, the command is:

PaperSizeSelect (Name: "B5 Landscape"; Width: 7.1658"; Height: 10.1183"; Type: Other!)

---

**See Also** ♦ PaperSizeAdd ♦ PaperSizeDelete

---

## ParagraphBorderDlg

---

**Purpose** ParagraphBorderDlg displays the Paragraph Border dialog box.

---

|                 |                                 |
|-----------------|---------------------------------|
| <b>Syntax</b>   | ParagraphBorderDlg ()           |
| <b>Route</b>    | Layout, Paragraph, Border\Fill. |
| <b>See Also</b> | ♦ ?ParagraphBorder              |

---

## ParagraphFormatDlg

---

|                |                                                              |
|----------------|--------------------------------------------------------------|
| <b>Purpose</b> | ParagraphFormatDlg displays the Paragraph Format dialog box. |
| <b>Syntax</b>  | ParagraphFormatDlg ()                                        |
| <b>Route</b>   | Layout, Paragraph, Format.                                   |

---

## ParagraphHangingIndent

---

|                |                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ParagraphHangingIndent aligns the first line of a paragraph with the left margin and indents the remaining lines to the next tab stop. |
| <b>Syntax</b>  | ParagraphHangingIndent ()                                                                                                              |
| <b>Route</b>   | Layout, Paragraph, Hanging Indent.                                                                                                     |

---

## ParagraphNumberDefDlg

---

|                |                                                                     |
|----------------|---------------------------------------------------------------------|
| <b>Purpose</b> | ParagraphNumberDefDlg displays the Change Outline Level dialog box. |
| <b>Syntax</b>  | ParagraphNumberDefDlg ()                                            |
| <b>Route</b>   | Tools, Outline, Options, Change Level.                              |

---

---

# ParagraphSpacing

|                   |                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ParagraphSpacing specifies the spacing of a Hard Return [Hrt], or the spacing between paragraphs. The default spacing is equal to the font point size plus 2 points. |
| <b>Syntax</b>     | ParagraphSpacing (AmtSp: <i>Numeric Expression</i> )                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                      |
| <i>AmtSp</i>      | Numeric Expression: Specifies spacing between paragraphs. The default is 1.0.                                                                                        |
| <b>Route</b>      | Layout, Paragraph, Format, specify a number in the Spacing Between Paragraphs entry box.                                                                             |
| <b>Example</b>    | To specify a spacing of 1.2 between paragraphs, the command is:<br><pre>ParagraphSpacing (AmtSp: AmtSp: 1.2)</pre>                                                   |
| <b>See Also</b>   | ♦ ?SpacingBelowPar                                                                                                                                                   |

---

# PasteLink

|                   |                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PasteLink inserts an object from the Clipboard into the current document, creating a link with the server application used to create the object.                       |
| <b>Syntax</b>     | PasteLink (FormatName: <i>Character Expression</i> ; FormatId: <i>Numeric Expression</i> ; LinkType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )             |
| <b>Parameters</b> |                                                                                                                                                                        |
| <i>FormatName</i> | Character Expression: Specifies the format of the linked object.                                                                                                       |
| <i>FormatId</i>   | Numeric Expression: Specifies an identification number for the format of the linked object. This number will vary according to the state of WordPerfect.               |
| <i>LinkType</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to create a DDE link or an OLE link.<br><ol style="list-style-type: none"><li>1 Dde!</li><li>2 Ole!</li></ol> |
| <b>Route</b>      | When the Clipboard contains an object created in another application, choose Edit, Paste Special, Paste Link.                                                          |

---

|                |                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | When the Clipboard contains a Paintbrush object, to link the object to the current WordPerfect document the command is:<br><br>PasteLink (FormatName: "P"; FormatId: 50585; LinkType: Ole!) |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## PasteSpecial

---

|                                   |                                                                                                                                                                                                                                               |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>                    | PasteSpecial specifies a data type when copying Clipboard contents into a document.<br>IMPORTANT: This command must be followed by EditPaste.                                                                                                 |
| <b>Syntax</b><br><i>PasteType</i> | PasteSpecial (PasteType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )<br>Numeric Equivalent: Specifies a data type.<br><ul style="list-style-type: none"><li>0 OnlySelection!</li><li>1 OnlyText!</li><li>2 OnlyFunctions!</li></ul> |
| <b>Route</b>                      | When the Clipboard is not empty, choose Edit, Paste Special, select a data type, Paste.                                                                                                                                                       |
| <b>Example</b>                    | To paste Clipboard contents when the Clipboard contains unformatted text, the command is:<br><br>PasteSpecial (PasteType: OnlyText!)<br>EditPaste (ClipboardFormat: "U"; FormatId: 1; LinkType: DDE!)                                         |
| <b>See Also</b>                   | ♦ EditPaste                                                                                                                                                                                                                                   |

---

## PauseCommand

---

|                                   |                                                                                                                                                                                   |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>                    | PauseCommand pauses macro execution until a specified command is encountered.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>                     | PauseCommand (Token: <i>Macro Command</i> )                                                                                                                                       |
| <b>Parameters</b><br><i>Token</i> | Macro Command: Specifies a macro command which pauses macro execution.                                                                                                            |

---

**Example**

To pause macro execution until a HardReturn is pressed, the command is:

PauseCommand (Token: HardReturn)

---

## PauseKey

---

**Purpose**

PauseKey pauses a macro until a specified key or character is pressed.

**Syntax**

PauseKey (Key: *Numeric Equivalent* or *Enumerated Type*, Character: *Character Expression*)

---

**Parameters***Key*

Numeric Equivalent: Specifies a key to resume macro execution. The default is Enter!.

- 0 Enter!
- 1 Cancel!
- 2 Close!
- 3 Character!
- 4 Any!

*Character*

Character Expression: Specifies a case-sensitive character that resumes macro execution when pressed. The Key parameter must be Character!.

---

**Route**

While recording a macro, choose Tools, Macro, Pause.

---

**Example**

To resume macro execution when Enter is pressed, the command is:

PauseKey (Key: Enter!)

---

## PauseSet

---

**Purpose**

PauseSet specifies a macro command to resume a paused macro.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax**

PauseSet (Token: *Macro Token*)

---



---

## Parameters

**Token** Macro Token: Specifies a macro token (command) to resume a paused macro. The macro token usually represents a keystroke.

---

## Example

To pause a macro until the Page Up key is pressed, the command is:

```
PauseSet (Token: PosScreenUp)
```

---

## See Also

♦ PAUSE

---

# PosCellDown

---

## Purpose

PosCellDown moves the insertion point down one row in the current table.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro. For a recordable command with the same function, use PosTableCellDown.

---

## Syntax

```
PosCellDown ()
```

---

## See Also

♦ PosCellNext ♦ PosCellPrevious ♦ PosCellUp ♦ PosTableCellDown

---

# PosCellNext

---

## Purpose

PosCellNext moves the insertion point one character to the right in the current table.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro. For a recordable command with the same function, use PosCharNext.

---

## Syntax

```
PosCellNext ()
```

---

## See Also

♦ PosCellDown ♦ PosCellPrevious ♦ PosCellUp ♦ PosCharNext

---

## PosCellPrevious

---

|                 |                                                                                                                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosCellPrevious moves the insertion point one character to the left in the current table.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command with the same function, use PosCharPrevious. |
| <b>Syntax</b>   | PosCellPrevious ()                                                                                                                                                                                                                                                      |
| <b>See Also</b> | ♦ PosCellDown ♦ PosCellNext ♦ PosCellUp ♦ PosCharPrevious                                                                                                                                                                                                               |

---

## PosCellUp

---

|                 |                                                                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosCellUp moves the insertion point up one row in the current table.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command with the same function, use PosTableCellUp. |
| <b>Syntax</b>   | PosCellUp ()                                                                                                                                                                                                                                      |
| <b>See Also</b> | ♦ PosCellDown ♦ PosCellNext ♦ PosCellPrevious                                                                                                                                                                                                     |

---

## PosCharacter

---

|                     |                                                                                                                                                                                                                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>      | PosCharacter searches forward and positions the insertion point after a specified character. This command is case sensitive.<br><br>IMPORTANT: This command only searches forward 2000 characters.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>       | PosCharacter (CharToMoveTo: <i>Character Expression</i> )                                                                                                                                                                                                                                                  |
| <b>Parameters</b>   |                                                                                                                                                                                                                                                                                                            |
| <i>CharToMoveTo</i> | Character Expression: Specifies a single search character.                                                                                                                                                                                                                                                 |

---

---

|                |                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To position the insertion point after the first C to the right of the insertion point, the command is:<br><br>PosCharacter (CharToMoveTo: "C") |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|

---

## PosCharNext

---

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| <b>Purpose</b> | PosCharNext moves the insertion point one character to the right. |
| <b>Syntax</b>  | PosCharNext ()                                                    |
| <b>Route</b>   | Press Right Arrow.                                                |

---

## PosCharPrevious

---

|                |                                                                      |
|----------------|----------------------------------------------------------------------|
| <b>Purpose</b> | PosCharPrevious moves the insertion point one character to the left. |
| <b>Syntax</b>  | PosCharPrevious ()                                                   |
| <b>Route</b>   | Press Left Arrow.                                                    |

---

## PosColBottom

---

|                |                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosColBottom positions the insertion point at the beginning of the last line of a current column. If the insertion point is not in columns, this command moves the insertion point to the beginning of the last line of the current page. |
| <b>Syntax</b>  | PosColBottom ()                                                                                                                                                                                                                           |
| <b>Route</b>   | Edit, Go To, Bottom of Column.                                                                                                                                                                                                            |

---

---

## PosColFirst

|                |                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosColFirst positions the insertion point at the left margin in the first column. If the insertion point is not in columns, this command moves the insertion point to the beginning of the current line. |
| <b>Syntax</b>  | PosColFirst ()                                                                                                                                                                                           |
| <b>Route</b>   | Edit, Go To, First Column.                                                                                                                                                                               |

---

## PosColLast

|                |                                                                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosColLast positions the insertion point in the last column of the current row. If the insertion point is not in columns, this command positions the insertion point at the beginning of the current line. |
| <b>Syntax</b>  | PosColLast ()                                                                                                                                                                                              |
| <b>Route</b>   | Edit, Go To, Last Column.                                                                                                                                                                                  |

---

## PosColNext

|                |                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosColNext moves the insertion point one column to the right. This command has no effect if the insertion point is not in columns. |
| <b>Syntax</b>  | PosColNext ()                                                                                                                      |
| <b>Route</b>   | Edit, Go To, Next Column.                                                                                                          |

---

## PosColPrevious

|                |                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosColPrevious moves the insertion point one column to the left. If the insertion point is not in columns, this command has no effect. |
| <b>Syntax</b>  | PosColPrevious ()                                                                                                                      |
| <b>Route</b>   | Edit, Go To, Previous Column.                                                                                                          |

---

## PosColTop

|                |                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosColTop moves the insertion point to the top of the current column. If the insertion point is not in columns, this command moves the insertion point to the beginning of the previous line. |
| <b>Syntax</b>  | PosColTop ()                                                                                                                                                                                  |
| <b>Route</b>   | Edit, Go To, Top of Column.                                                                                                                                                                   |

---

## PosDocBottom

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| <b>Purpose</b> | PosDocBottom moves the insertion point to the end of the current document. |
| <b>Syntax</b>  | PosDocBottom ()                                                            |
| <b>Route</b>   | Press Ctrl+End.                                                            |

---

## PosDocTop

|                |                                                                        |
|----------------|------------------------------------------------------------------------|
| <b>Purpose</b> | PosDocTop moves the insertion point to the beginning of document text. |
| <b>Syntax</b>  | PosDocTop ()                                                           |
| <b>Route</b>   | Press Ctrl+Home.                                                       |

---

## PosDocVeryTop

---

|                |                                                                                      |
|----------------|--------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosDocVeryTop moves the insertion point to the beginning of a document before codes. |
| <b>Syntax</b>  | PosDocVeryTop ()                                                                     |
| <b>Route</b>   | Press Ctrl+Home, Ctrl+Home.                                                          |

---

## PosFunction

---

|                |                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosFunction moves the insertion point to the right of the next occurrence of the code specified in the Function parameter.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| <b>Syntax</b> | PosFunction (Function: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|------------------------------------------------------------------------------|

---

### Parameters

|                 |                                                                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Function</i> | Numeric Equivalent or Enumerated Type: Specifies a code.<br><br>129 HardSpace!<br>132 HardHyphen!<br>144 CancelHyphenation!<br>204 HardEndOfLine!<br>207 SoftEndOfLine!<br>224 Tab! |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                |                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To move the insertion point to the right of a Hard Space, the command is:<br><br>PosFunction (Function: HardSpace!) |
|----------------|---------------------------------------------------------------------------------------------------------------------|

|                 |                |
|-----------------|----------------|
| <b>See Also</b> | ♦ PosCharacter |
|-----------------|----------------|

---

## PosGoPrevious

---

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| <b>Purpose</b> | PosGoPrevious moves the insertion point to its previous position. |
|----------------|-------------------------------------------------------------------|

---

|               |                             |
|---------------|-----------------------------|
| <b>Syntax</b> | PosGoPrevious ()            |
| <b>Route</b>  | Edit, Go To, Last Position. |

---

## PosGoToDlg

---

|                |                                           |
|----------------|-------------------------------------------|
| <b>Purpose</b> | PosGoToDlg displays the Go To dialog box. |
| <b>Syntax</b>  | PosGoToDlg ()                             |
| <b>Route</b>   | Edit, Go To.                              |

---

## PosLineBeg

---

|                |                                                                                    |
|----------------|------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosLineBeg moves the insertion point to the beginning of the current line of text. |
| <b>Syntax</b>  | PosLineBeg ()                                                                      |
| <b>Route</b>   | Press Home.                                                                        |

---

## PosLineDown

---

|                |                                                      |
|----------------|------------------------------------------------------|
| <b>Purpose</b> | PosLineDown moves the insertion point down one line. |
| <b>Syntax</b>  | PosLineDown ()                                       |
| <b>Route</b>   | Press Down Arrow.                                    |

---

## PosLineEnd

---

|                |                                                                              |
|----------------|------------------------------------------------------------------------------|
| <b>Purpose</b> | PosLineEnd moves the insertion point to the end of the current line of text. |
| <b>Syntax</b>  | PosLineEnd ()                                                                |

---

---

|              |            |
|--------------|------------|
| <b>Route</b> | Press End. |
|--------------|------------|

---

## PosLineUp

---

|                |                                                  |
|----------------|--------------------------------------------------|
| <b>Purpose</b> | PosLineUp moves the insertion point up one line. |
|----------------|--------------------------------------------------|

---

|               |              |
|---------------|--------------|
| <b>Syntax</b> | PosLineUp () |
|---------------|--------------|

---

|              |                 |
|--------------|-----------------|
| <b>Route</b> | Press Up Arrow. |
|--------------|-----------------|

---

## PosLineVeryBeg

---

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| <b>Purpose</b> | PosLineVeryBeg moves the insertion point to the beginning of a line before codes. |
|----------------|-----------------------------------------------------------------------------------|

---

|               |                   |
|---------------|-------------------|
| <b>Syntax</b> | PosLineVeryBeg () |
|---------------|-------------------|

---

|              |                   |
|--------------|-------------------|
| <b>Route</b> | Press Home, Home. |
|--------------|-------------------|

---

## PosLineVeryEnd

---

|                |                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosLineVeryEnd moves the insertion point to the end of a line after codes (with the exception of end of line codes). |
|----------------|----------------------------------------------------------------------------------------------------------------------|

---

|               |                   |
|---------------|-------------------|
| <b>Syntax</b> | PosLineVeryEnd () |
|---------------|-------------------|

---

|              |                 |
|--------------|-----------------|
| <b>Route</b> | Press End, End. |
|--------------|-----------------|

---

## PosPage

---

|                |                                                        |
|----------------|--------------------------------------------------------|
| <b>Purpose</b> | PosPage moves the insertion point to a specified page. |
|----------------|--------------------------------------------------------|

---

|               |                                                                                                                                                                                                                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | PosPage (Volume: <i>Numeric Expression</i> or <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Chapter: <i>Numeric Expression</i> or <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Page: <i>Numeric Expression</i> ; Direction: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



---

**Parameters**

|                  |                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Volume</i>    | Numeric Expression or Numeric Equivalent or Enumerated Type: Specifies a volume number. Use the numeric equivalent or enumerated type if no volume number is needed.<br>0 NoneSpecified!   |
| <i>Chapter</i>   | Numeric Expression or Numeric Equivalent or Enumerated Type: Specifies a chapter number. Use the numeric equivalent or enumerated type if no chapter number is needed.<br>0 NoneSpecified! |
| <i>Page</i>      | Numeric Expression: Specifies a page number.                                                                                                                                               |
| <i>Direction</i> | Numeric Equivalent or Enumerated Type: Specifies whether the page is up or down.<br>0 Down!<br>1 Up!                                                                                       |

---

**Route** Edit, Go To, specify a page number.

---

**Example** To position the insertion point at page three of chapter two in volume one, the command is:

PosPage (Volume: 1; Chapter: 2; Page: 3; Direction: Up!)

---

## PosPageBottom

---

**Purpose** PosPageBottom moves the insertion point to the bottom of the current page.

---

**Syntax** PosPageBottom ()

---

**Route** Edit, Go To, Bottom of Current Page.

---

## PosPageNext

---

**Purpose** PosPageNext moves the insertion point to the beginning of the next page. If the insertion point is on the last page of the document, this command moves the insertion point to the end of the page.

---

|               |                      |
|---------------|----------------------|
| <b>Syntax</b> | PosPageNext ()       |
| <b>Route</b>  | Press Alt+Page Down. |

---

## PosPagePrevious

---

|                |                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosPagePrevious moves the insertion point to the beginning of the previous page. If the insertion point is on the first page of the document, this command moves the insertion point to the top of that page. |
| <b>Syntax</b>  | PosPagePrevious ()                                                                                                                                                                                            |
| <b>Route</b>   | Press Alt+Page Up.                                                                                                                                                                                            |

---

## PosPageTop

---

|                |                                                                      |
|----------------|----------------------------------------------------------------------|
| <b>Purpose</b> | PosPageTop moves the insertion point to the top of the current page. |
| <b>Syntax</b>  | PosPageTop ()                                                        |
| <b>Route</b>   | Press Alt+Home.                                                      |

---

## PosParagraphNext

---

|                 |                                                                                    |
|-----------------|------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosParagraphNext moves the insertion point to the beginning of the next paragraph. |
| <b>Syntax</b>   | PosParagraphNext ()                                                                |
| <b>Route</b>    | Press Ctrl+Down Arrow.                                                             |
| <b>See Also</b> | ♦ PosParagraphPrevious                                                             |

---

---

## PosParagraphPrevious

|                 |                                                                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosParagraphPrevious moves the insertion point to the beginning of the current paragraph. If the insertion point is at the beginning of a paragraph, this command moves the insertion point to the beginning of the previous paragraph. |
| <b>Syntax</b>   | PosParagraphPrevious ()                                                                                                                                                                                                                 |
| <b>Route</b>    | Press Ctrl+Up Arrow.                                                                                                                                                                                                                    |
| <b>See Also</b> | ♦ PosParagraphNext                                                                                                                                                                                                                      |

---

## PosScreenDown

|                |                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosScreenDown moves the insertion point to the bottom of the screen, then moves the insertion point down one screen at a time with each repetition of the command. |
| <b>Syntax</b>  | PosScreenDown ()                                                                                                                                                   |
| <b>Route</b>   | Press Page Down.                                                                                                                                                   |

---

## PosScreenLeft

|                 |                                                                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosScreenLeft moves the insertion point to the left edge of the screen, then moves the insertion point to the left one screen at a time with each repetition of the command.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | PosScreenLeft ()                                                                                                                                                                                                                                                                     |
| <b>See Also</b> | ♦ SelectScreenLeft                                                                                                                                                                                                                                                                   |

---

## PosScreenRight

|                |                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosScreenRight moves the insertion point to the right edge of the screen, then moves the insertion point to the right one screen at a time with each repetition of the command. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PosScreenRight ()

---

**See Also** ♦ SelectScreenRight

---

## PosScreenUp

---

**Purpose** PosScreenUp moves the insertion point to the top of the screen, then moves the insertion point up one screen at a time with each repetition of the command.

---

**Syntax** PosScreenUp ()

---

**Route** Press Page Up.

---

## PosSelectBottom

---

**Purpose** PosSelectBottom moves the insertion point to the end (bottom) of the selected text.

---

**Syntax** PosSelectBottom ()

---

**Route** Edit, Go To, End of Selection.

---

## PosSelectTop

---

**Purpose** PosSelectTop positions the insertion point at the beginning of the selected text.

---

**Syntax** PosSelectTop ()

---

**Route** Edit, Go To, Beginning of Selection.

---

## PosTableBegin

|                 |                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosTableBegin positions the insertion point in the first cell of a table.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | PosTableBegin ()                                                                                                                                                                                                 |
| <b>Route</b>    | Edit, Go To, Position, First Cell.                                                                                                                                                                               |
| <b>See Also</b> | ♦ PosTableEnd                                                                                                                                                                                                    |

---

## PosTableBlockBeg

|                |                                                                                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableBlockBeg moves the insertion point to the first cell of a selection in a table.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableBlockBeg ()                                                                                                                                                                                                            |
| <b>Route</b>   | Edit, Go To, Position, Beginning of Selection.                                                                                                                                                                                 |

---

## PosTableCell

|                   |                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PosTableCell moves the insertion point to the cell specified in the Cell parameter.                         |
| <b>Syntax</b>     | PosTableCell (Cell: <i>Character Expression</i> )                                                           |
| <b>Parameters</b> |                                                                                                             |
| <i>Cell</i>       | Character Expression: Specifies a table and table cell to move the insertion point to.                      |
| <b>Route</b>      | Edit, Go To, Table, specify a table, Cell/Range, specify a cell.                                            |
| <b>Example</b>    | To position the insertion point in cell B3 of Table A, the command is:<br>PosTableCell (Cell: "Table A.B3") |

---

## PosTableCellBottom

---

|                |                                                                                                                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableCellBottom positions the insertion point at the beginning of the last line in a table cell.<br><br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableCellBottom ()                                                                                                                                                                                                                          |
| <b>Route</b>   | Edit, Go To, Position, select Bottom of Cell.                                                                                                                                                                                                  |

---

## PosTableCellDown

---

|                |                                                                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableCellDown moves the insertion point down one cell in a table.<br><br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableCellDown ()                                                                                                                                                                                             |
| <b>Route</b>   | Edit, Go To, Position, Next Row.                                                                                                                                                                                |

---

## PosTableCellNext

---

|                |                                                                                                                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableCellNext moves the insertion point one cell to the right.<br><br>IMPORTANT: If the insertion point is in the far-right cell, the insertion point moves to the first cell in the next row. If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableCellNext ()                                                                                                                                                                                                                                                                                                        |
| <b>Route</b>   | Edit, Go To, Position, Next Column.                                                                                                                                                                                                                                                                                        |

---

## PosTableCellPrevious

|                |                                                                                                                                                                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableCellPrevious moves the insertion point one cell to the left.<br><br>IMPORTANT: If the insertion point is in the far-left cell, the insertion point moves to the last cell on the previous row. If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableCellPrevious ()                                                                                                                                                                                                                                                                                                         |
| <b>Route</b>   | Edit, Go To, Position, Previous Column.                                                                                                                                                                                                                                                                                         |

---

## PosTableCellTop

|                |                                                                                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableCellTop positions the insertion point at the top of the current table cell.<br><br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableCellTop ()                                                                                                                                                                                                             |
| <b>Route</b>   | Edit, Go To, Position, Top of Cell.                                                                                                                                                                                            |

---

## PosTableCellUp

|                |                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableCellUp moves the insertion point up one table cell.<br><br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableCellUp ()                                                                                                                                                                                      |
| <b>Route</b>   | Edit, Go To, Position, Previous Row.                                                                                                                                                                   |

---

## PosTableColBottom

|                |                                                                                                                                                                                                                             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableColBottom moves the insertion point to the bottom of the current column.<br><br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                                         |
|---------------|-----------------------------------------|
| <b>Syntax</b> | PosTableColBottom ()                    |
| <b>Route</b>  | Edit, Go To, Position, Bottom of Column |

---

## PosTableColumnTop

---

|                |                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableColumnTop moves the insertion point to the top of the current column.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableColumnTop ()                                                                                                                                                                                                 |
| <b>Route</b>   | Edit, Go To, Position, Top of Column.                                                                                                                                                                                |

---

## PosTableEnd

---

|                 |                                                                              |
|-----------------|------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosTableEnd moves the insertion point to the last cell of the current table. |
| <b>Syntax</b>   | PosTableEnd ()                                                               |
| <b>Route</b>    | Layout, Go To, Position, Last Cell.                                          |
| <b>See Also</b> | ♦ PosTableBegin ♦ PosTableRowBegin ♦ PosTableRowEnd                          |

---

## PosTableGoPrevious

---

|                |                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableGoPrevious moves the insertion point to its previous position in a table.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableGoPrevious ()                                                                                                                                                                                                                                                                                                        |

---



---

## PosTableRowBegin

|                 |                                                                                                                                                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosTableRowBegin moves the insertion point to the first cell of a current row.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | PosTableRowBegin ()                                                                                                                                                                                                   |
| <b>Route</b>    | Edit, Go To, Position, First Column.                                                                                                                                                                                  |
| <b>See Also</b> | ♦ PosTableEnd                                                                                                                                                                                                         |

---

## PosTableRowEnd

|                 |                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PosTableRowEnd moves the insertion point to the last cell of the current row.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | PosTableRowEnd ()                                                                                                                                                                                                    |
| <b>Route</b>    | Edit, Go To, Position, Last Column.                                                                                                                                                                                  |
| <b>See Also</b> | ♦ PosTableEnd                                                                                                                                                                                                        |

---

## PosTableTopCellNext

|                |                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableTopCellNext moves the insertion point one cell to the right.<br>IMPORTANT: If the insertion point is in the last cell of a table, this command creates a new table row at the bottom of the table. If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableTopCellNext ()                                                                                                                                                                                                                                                                                                              |
| <b>Route</b>   | Press Tab while the insertion point is in a table.                                                                                                                                                                                                                                                                                  |

---

## PosTableTopCellPrevious

|                |                                                                                                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PosTableTopCellPrevious moves the insertion point one cell to the left.<br>IMPORTANT: If the insertion point is not in a table, this command ends the macro or redirects it to the LABEL specified by ONERROR. |
| <b>Syntax</b>  | PosTableTopCellPrevious ()                                                                                                                                                                                     |
| <b>Route</b>   | Press Shift+Tab while the insertion point is in a table.                                                                                                                                                       |

---

## PosWordNext

|                |                                                                          |
|----------------|--------------------------------------------------------------------------|
| <b>Purpose</b> | PosWordNext moves the insertion point to the beginning of the next word. |
| <b>Syntax</b>  | PosWordNext ()                                                           |
| <b>Route</b>   | Press Ctrl+Right Arrow.                                                  |

---

## PosWordPrevious

|                |                                                                                  |
|----------------|----------------------------------------------------------------------------------|
| <b>Purpose</b> | PosWordPrevious moves the insertion point to the beginning of the previous word. |
| <b>Syntax</b>  | PosWordPrevious ()                                                               |
| <b>Route</b>   | Press Ctrl+Left Arrow.                                                           |

---

## PowerBarShow

|                |                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PowerBarShow specifies whether to hide or show the Power Bar. If no parameter is specified, this command acts as a toggle. |
| <b>Syntax</b>  | PowerBarShow (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                 |

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether Power Bar display is on or off.

0 Off!  
1 On!

---

**Route** View, Power Bar.

---

**Example** To display the Power Bar, the command is:

PowerBarShow (State: On!)

---

## PrefBackupDlg

---

**Purpose** PrefBackupDlg displays the File Preferences dialog box with the Documents/Backup radio button selected.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PrefBackupDlg ()

---

## PrefBackupPath

---

**Purpose** PrefBackupPath specifies the document backup directory and whether to update the QuickList.

---

**Syntax** PrefBackupPath (BackupDir: *Character Expression*; UpdateQuickList: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*BackupDir* Character Expression: Specifies the document backup directory.

*UpdateQuickList* Numeric Equivalent or Enumerated Type: Specifies whether to update the QuickList.

0 No!  
1 Yes!

---

**Route** File, Preferences, File, Documents/Backup, specify a backup directory, select or deselect Update QuickList with Changes.

---

**Example** To specify C:\WPWIN60\FILES\ as the backup directory and to update the quicklist, the command is:

```
PrefBackupPath (BackupDir: "C:\WPWIN60\FILES\"; UpdateQuickList: Yes!)
```

---

## PrefBeep

---

**Purpose** PrefBeep specifies when the computer beeps.

---

**Syntax** PrefBeep (Error: *Numeric Equivalent* or *Enumerated Type*; Hyphenation: *Numeric Equivalent* or *Enumerated Type*; SearchFailure: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Error* Numeric Equivalent or Enumerated Type: Specifies whether a beep occurs on errors.

|   |      |
|---|------|
| 0 | Off! |
| 1 | On!  |

*Hyphenation* Numeric Equivalent or Enumerated Type: Specifies whether a beep occurs when the user is required to position a hyphen.

|   |      |
|---|------|
| 0 | Off! |
| 1 | On!  |

*SearchFailure* Numeric Equivalent or Enumerated Type: Specifies whether a beep occurs when a search fails.

|   |      |
|---|------|
| 0 | Off! |
| 1 | On!  |

---

**Route** File, Preferences, Environment, Beep On, select options.

---

**Example** To set beep options, the command is:

```
PrefBeep (Error: On!; Hyphenation: On!; SearchFailure: Off!)
```

---

**See Also** ♦ PrefEnvSettingsDlg ♦ PrefSave

---

## PrefButtonBar

---

|                |                                                                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PrefButtonBar displays the Button Bar Preferences dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | PrefButtonBar ()                                                                                                                                                      |

---

## PrefDisplayDlg

---

|                |                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PrefDisplayDlg displays the Display Preferences dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | PrefDisplayDlg ()                                                                                                                                                   |

---

## PrefDisplaySet

---

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PrefDisplaySet changes the way features display.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Syntax</b>  | PrefDisplaySet (DefaultView: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , ShowTableGridLines: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , ShowComments: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , ShowGraphics: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , TextInSystemColors: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , ShowHiddenText: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , SculpturedDialogs: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , VerticalScrollBar: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , HorizontalScrollBar: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

### Parameters

|                           |                                                                                                                                                                             |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>DefaultView</i>        | Numeric Equivalent or Enumerated Type: Specifies a default view.<br><ul style="list-style-type: none"><li>0 ViewDraft!</li><li>1 ViewPage!</li><li>2 ViewTwoPage!</li></ul> |
| <i>ShowTableGridLines</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display table grid lines.<br><ul style="list-style-type: none"><li>0 No!</li><li>1 Yes!</li></ul>               |

|                            |                                                                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <i>ShowComments</i>        | Numeric Equivalent or Enumerated Type: Specifies whether to display comments.<br>0 No!<br>1 Yes!                                    |
| <i>ShowGraphics</i>        | Numeric Equivalent or Enumerated Type: Specifies whether to display graphics.<br>0 No!<br>1 Yes!                                    |
| <i>TextInSystemColors</i>  | Numeric Equivalent or Enumerated Type: Specifies whether to display text with system colors.<br>0 No!<br>1 Yes!                     |
| <i>ShowHiddenText</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to display hidden text.<br>0 No!<br>1 Yes!                                 |
| <i>SculpturedDialogs</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to display sculptured dialog boxes.<br>0 No!<br>1 Yes!                     |
| <i>VerticalScrollBar</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to display the vertical scroll bar.<br>0 No!<br>1 Yes!                     |
| <i>HorizontalScrollBar</i> | Numeric Equivalent or Enumerated Type: Specifies how to display the horizontal scroll bar.<br>0 Hide!<br>1 Show!<br>2 WhenRequired! |

---

**Route** File, Preferences, Display, View/Zoom, select a default view, Document, select options.

---

**Example** To set display options, the command is:

```
PrefDisplaySet (DefaultView: ViewPage!; ShowTableGridLines: No!; ShowComments: No!; ShowGraphics: No!; TextInSystemColors: Yes!; ShowHiddenText: No!; SculpturedDialogs: Yes!; VerticalScrollBar: No!; HorizontalScrollBar: Show!)
```

---

# PrefDocSummary

---

|                               |                                                                                                                                                                                                                      |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>                | PrefDocSummary specifies a subject search text and default descriptive type for a document.                                                                                                                          |
| <b>Syntax</b>                 | PrefDocSummary (SubjectSearchText: <i>Character Expression</i> ; DefaultDescriptiveType: <i>Character Expression</i> ; CreateOnExit: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                           |
| <b>Parameters</b>             |                                                                                                                                                                                                                      |
| <i>SubjectSearchText</i>      | Character Expression: Specifies text that precedes a document's subject line.                                                                                                                                        |
| <i>DefaultDescriptiveType</i> | Character Expression: Specifies default text to display in the Descriptive Type entry box.                                                                                                                           |
| <i>CreateOnExit</i>           | Numeric Equivalent or Enumerated Type: Specifies whether a document summary is created when documents are saved or closed.<br><br>0 No!<br>1 Yes!                                                                    |
| <b>Route</b>                  | File, Preferences, Summary, specify options.                                                                                                                                                                         |
| <b>Example</b>                | To create Document Summary subject search text and a default description, the command is:<br><br>PrefDocSummary (SubjectSearchText: "Research Paper"; DefaultDescriptiveType: "WordPerfect 6.0"; CreateOnExit: Yes!) |
| <b>See Also</b>               | ♦ PrefDocSummaryDlg ♦ PrefSave                                                                                                                                                                                       |

---

# PrefDocSummaryDlg

---

|                 |                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | PrefDocSummaryDlg displays the Document Summary Preferences dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | PrefDocSummaryDlg ()                                                                                                                                                            |
| <b>See Also</b> | ♦ PrefDocSummary                                                                                                                                                                |

---

# PrefEnvironment

---

**Purpose** PrefEnvironment specifies the default environment settings.

**Syntax** PrefEnvironment (UserName: *Character Expression*; UserInitials: *Character Expression*; UserColorRed: *Numeric Expression*; UserColorGreen: *Numeric Expression*; UserColorBlue: *Numeric Expression*; ConfirmCodeDeletion: *Numeric Equivalent or Enumerated Type*; ConfirmTableFormulaDeletion: *Numeric Equivalent or Enumerated Type*; SetQuickMarkOnSave: *Numeric Equivalent or Enumerated Type*; ActivateHypertext: *Numeric Equivalent or Enumerated Type*; FormatForDefaultPrinter: *Numeric Equivalent or Enumerated Type*; SaveWorkSpace: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

|                                    |                                                                                                                        |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <i>UserName</i>                    | Character Expression: Specifies a user name.                                                                           |
| <i>UserInitials</i>                | Character Expression: Specifies a user ID.                                                                             |
| <i>UserColorRed</i>                | Numeric Expression: Specifies the amount of red in a user color. Values range from 0 to 255.                           |
| <i>UserColorGreen</i>              | Numeric Expression: Specifies the amount of green in a user color. Values range from 0 to 255.                         |
| <i>UserColorBlue</i>               | Numeric Expression: Specifies the amount of blue in a user color. Values range from 0 to 255.                          |
| <i>ConfirmCodeDeletion</i>         | Numeric Equivalent or Enumerated Type: Specifies whether to confirm code deletion.<br>0 No!<br>1 Yes!                  |
| <i>ConfirmTableFormulaDeletion</i> | Numeric Equivalent or Enumerated Type: Specifies whether to confirm table formula deletion.<br>0 No!<br>1 Yes!         |
| <i>SetQuickMarkOnSave</i>          | Numeric Equivalent or Enumerated Type: Specifies whether to set a QuickMark when saving a document.<br>0 No!<br>1 Yes! |
| <i>ActivateHypertext</i>           | Numeric Equivalent or Enumerated Type: Specifies whether to activate Fast Save.<br>0 Off!<br>1 On!                     |



*FormatForDefault-Printer*

Numeric Equivalent or Enumerated Type: Specifies whether to format documents for the default printer on retrieval.

- 0 No!
- 1 Yes!

*SaveWorkSpace*

Numeric Equivalent or Enumerated Type: Specifies whether to save the current WorkSpace on exit.

- 0 Always!
- 1 Never!
- 2 PromptOnExit!

---

**Route** File, Preferences, Environment, specify options.

---

**Example** To specify environment settings, the command is:

```
PrefEnvironment (UserName: "John Doe"; UserInitials: "JZD"; UserColorRed: 255; UserColorGreen: 0; UserColorBlue: 0; ConfirmCodeDeletion: Yes!; ConfirmFormulaDeletion: No!; SetQuickMarkOnSave: No!; ActivateHypertext: Off!; FormatForDefaultPrinter: Yes!; SaveWorkSpace: PromptOnExit!)
```

---

## PrefEnvSettingsDlg

---

**Purpose** PrefEnvSettingsDlg displays the Environment Preferences dialog box.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PrefEnvSettingsDlg ()

---

## PrefHyphenation

---

**Purpose** PrefHyphenation specifies whether Hyphenation is on, and how it performs.

---

**Syntax** PrefHyphenation (Prompt: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*Prompt* Numeric Equivalent or Enumerated Type: Specifies if and when to prompt for hyphen placement.

- 0 Always!
- 1 Never!
- 2 WhenRequired!

|                 |                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------|
| <b>Route</b>    | File, Preferences, Environment, Formatting, select a hyphenation prompt.                    |
| <b>Example</b>  | To set hyphenation options, the command is:<br><pre>PrefHyphenation (Prompt: Always!)</pre> |
| <b>See Also</b> | ♦ PrefEnvSettingsDlg ♦ PrefSave                                                             |

## PrefImportDlg

|                |                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PrefImportDlg displays the Import Preferences dialog box.<br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | PrefImportDlg ()                                                                                                                                                     |

## PrefImportOptions

|                               |                                                                                                                                                                                                                       |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>                | PrefImportOptions specifies merge delimiters.                                                                                                                                                                         |
| <b>Syntax</b>                 | PrefImportOptions (FieldDelimiters: <i>Character Expression</i> ; RecordDelimiters: <i>Character Expression</i> ; StripCharacter: <i>Character Expression</i> ; EncapsulationCharacter: <i>Character Expression</i> ) |
| <b>Parameters</b>             |                                                                                                                                                                                                                       |
| <i>FieldDelimiters</i>        | Character Expression: Specifies a delimiter to separate fields.                                                                                                                                                       |
| <i>RecordDelimiters</i>       | Character Expression: Specifies a delimiter to separate records.                                                                                                                                                      |
| <i>StripCharacter</i>         | Character Expression: Specifies a character to strip during conversion.                                                                                                                                               |
| <i>EncapsulationCharacter</i> | Character Expression: Specifies a delimiter to protect information during conversion.                                                                                                                                 |
| <b>Route</b>                  | File, Preferences, Import, specify delimiters.                                                                                                                                                                        |
| <b>Example</b>                | To set import options, the command is:<br><pre>PrefImportOptions (FieldDelimiters: "%"; RecordDelimiters: "@"; StripCharacter: "!"; EncapsulationCharacter: "/")</pre>                                                |

# PrefLiteCodes

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrefLiteCodes specifies whether to display specified page formatting codes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Syntax</b>     | PrefLiteCodes (Show: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Space: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; HardReturn: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Tab: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Indent: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; SoftHyphen: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Advance: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Center: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; CenterPage: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; FlushRight: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>Show</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to display symbols on new and current documents.<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <i>Space</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to display the space symbol.<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>HardReturn</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display the hard return symbol.<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>Tab</i>        | Numeric Equivalent or Enumerated Type: Specifies whether to display the tab symbol.<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>Indent</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to display the indent symbol.<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>SoftHyphen</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display the soft hyphen symbol.<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

|                   |                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------|
| <i>Advance</i>    | Numeric Equivalent or Enumerated Type: Specifies whether to display the advance symbol.<br>0 No!<br>1 Yes!     |
| <i>Center</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to display the center symbol.<br>0 No!<br>1 Yes!      |
| <i>CenterPage</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display the center page symbol.<br>0 No!<br>1 Yes! |
| <i>FlushRight</i> | Numeric Equivalent or Enumerated Type: Specifies whether to display the flush right symbol.<br>0 No!<br>1 Yes! |

---

**Route** File, Preferences, Display, Show, select options.

---

**Example** To display hard return codes in new and current documents, the command is:  
PrefLiteCodes (Show: Yes!; Space: No!; HardReturn: Yes!; Tab: No!; Indent: No!; SoftHyphen: No!; Advance: No!; Center: No!; CenterPage: No!; FlushRight: No!)

---

## PrefLocationDlg

---

**Purpose** PrefLocationDlg displays the File Preferences dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PrefLocationDlg ()

---

## PrefLocationOfDocuments

---

**Purpose** PrefLocationOfDocuments specifies the location of document and backup files.

---

**Syntax** PrefLocationOfDocuments (Path: *Character Expression*; UseDefExt: *Numeric Equivalent* or *Enumerated Type*; DefExt: *Character Expression*; UpdateQuickList: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Path* Character Expression: Specifies the default document and backup directory.

*UseDefExt* Numeric Equivalent or Enumerated Type: Specifies whether to use the default extension on Open and Save.

0 No!  
1 Yes!

*DefExt* Character Expression: Specifies a default filename extension.

*UpdateQuickList* Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes.

0 No!  
1 Yes!

---

**Route** File, Preferences, File, Documents/Backup, specify options.

---

**Example** To specify the location of document and backup files, the command is:

PrefLocationOfDocuments (Path: "C:\WPWIN60\DOCSV"; UseDefExt: Yes!; DefExt: ".DOC"; UpdateQuickList: Yes!)

---

## PrefLocationOfGraphicsFiles

---

**Purpose** PrefLocationOfGraphicsFiles specifies the location of graphics files.

---

**Syntax** PrefLocationOfGraphicsFiles (GraphicsDir: *Character Expression*; GraphicsSupplDir: *Character Expression*; UpdateQuickList: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*GraphicsDir* Character Expression: Specifies the graphics default directory.

*GraphicsSupplDir* Character Expression: Specifies the graphics supplemental directory.

*UpdateQuickList* Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes.

0 No!  
1 Yes!

---

|                |                                                                                                                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | File, Preferences, File, Graphics, specify options.                                                                                                                                                     |
| <b>Example</b> | To specify the location of graphics files, the command is:<br><br>PrefLocationOfGraphicsFiles (GraphicsDir: "C:\WPWIN60\GRAPHICS"; GraphicsSupplDir: "C:\WPWIN60\GRAPHICS\SUPP"; UpdateQuickList: Yes!) |

---

## PrefLocationOfHyphenationFiles

---

|                        |                                                                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>         | PrefLocationOfHyphenationFiles specifies the location of hyphenation files.                                                                          |
| <b>Syntax</b>          | PrefLocationOfHyphenationFiles (HyphenationDir: <i>Character Expression</i> ; UpdateQuickList: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b>      |                                                                                                                                                      |
| <i>HyphenationDir</i>  | Character Expression: Specifies the default hyphenation directory.                                                                                   |
| <i>UpdateQuickList</i> | Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes.<br><br>0 No!<br>1 Yes!                                    |

---

|                |                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | File, Preferences, File, Hyphenation, specify options.                                                                                                           |
| <b>Example</b> | To specify the location of hyphenation files, the command is:<br><br>PrefLocationOfHyphenationFiles (HyphenationDir: "C:\WPWIN60\HYPHEN"; UpdateQuickList: Yes!) |

---

## PrefLocationOfMacroFiles

---

|                   |                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrefLocationOfMacroFiles specifies the location of macro files.                                                                                                                       |
| <b>Syntax</b>     | PrefLocationOfMacroFiles (MacroDir: <i>Character Expression</i> ; MacroSupplDir: <i>Character Expression</i> ; UpdateQuickList: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                                                                                                                       |
| <i>MacroDir</i>   | Character Expression: Specifies the default macro directory.                                                                                                                          |

---

|                        |                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------|
| <i>MacroSupplDir</i>   | Character Expression: Specifies the default macro supplemental directory.                  |
| <i>UpdateQuickList</i> | Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes. |
|                        | 0 No!                                                                                      |
|                        | 1 Yes!                                                                                     |

---

**Route** File, Preferences, File, Macros, specify options.

---

**Example** To specify the location of macro files, the command is:

PrefLocationOfMacroFiles (MacroDir: "C:\WPWIN60\MACROSI"; MacroSupplDir: "C:\WPWIN60\MACROSI\SUPP"; UpdateQuickList: Yes!)

---

## PrefLocationOfPrinterFiles

---

**Purpose** PrefLocationOfPrinterFiles specifies the location of printer files.

---

**Syntax** PrefLocationOfPrinterFiles (PrinterDir: *Character Expression*; PrinterSupplDir: *Character Expression*; LabelFile: *Character Expression*; LabelsDisplay: *Numeric Equivalent or Enumerated Type*; UpdateQuickList: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

|                        |                                                                             |
|------------------------|-----------------------------------------------------------------------------|
| <i>PrinterDir</i>      | Character Expression: Specifies the default printer directory.              |
| <i>PrinterSupplDir</i> | Character Expression: Specifies the default printer supplemental directory. |
| <i>LabelFile</i>       | Character Expression: Specifies the default label file directory.           |
| <i>LabelsDisplay</i>   | Numeric Equivalent or Enumerated Type: Specifies default labels to display. |
|                        | 0 DisplayLaser!                                                             |
|                        | 1 DisplayTract!                                                             |
|                        | 2 DisplayBoth!                                                              |

|                        |                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------|
| <i>UpdateQuickList</i> | Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes. |
|                        | 0 No!                                                                                      |
|                        | 1 Yes!                                                                                     |

---

**Route** File, Preferences, File, Printers/Labels, specify options.

---

**Example**

To specify the location of printer and label files, the command is:

```
PrefLocationOfPrinterFiles (PrinterDir: "C:\WPWIN60\PRINTER"; PrinterSupplDir:
"C:\WPWIN60\PRINTER\SUPP"; LabelFile: "C:\WPWIN60\LABEL"; LabelsDisplay: Display Both;
UpdateQuickList: Yes!)
```

---

## PrefLocationOfSpreadsheetFiles

---

**Purpose**

PrefLocationOfSpreadsheetFiles specifies the location of spreadsheet files.

---

**Syntax**

PrefLocationOfSpreadsheetFiles (SpreadsheetDir: *Character Expression*;  
SpreadsheetSupplDir: *Character Expression*; DatabaseDir: *Character Expression*;  
DatabaseSupplDir: *Character Expression*; UpdateQuickList: *Numeric Equivalent or  
Enumerated Type*)

---

**Parameters**

|                            |                                                                                                                   |
|----------------------------|-------------------------------------------------------------------------------------------------------------------|
| <i>SpreadsheetDir</i>      | Character Expression: Specifies the default spreadsheet directory.                                                |
| <i>SpreadsheetSupplDir</i> | Character Expression: Specifies the default supplemental spreadsheet directory.                                   |
| <i>DatabaseDir</i>         | Character Expression: Specifies the default database directory.                                                   |
| <i>DatabaseSupplDir</i>    | Character Expression: Specifies the default supplemental database directory.                                      |
| <i>UpdateQuickList</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes.<br><br>0 No!<br>1 Yes! |

---

**Route**

File, Preferences, File, Spreadsheets, specify options.

---

**Example**

To specify the location of spreadsheet files, the command is:

```
PrefLocationOfSpreadsheetFiles (SpreadsheetDir: "C:\WPWIN60\SPREAD"; SpreadsheetSupplDir:
"C:\WPWIN60\SPREAD\SUPP"; DatabaseDir: "C:\WPWIN60\DATABAS"; DatabaseSupplDir:
"C:\WPWIN60\DATABAS\SUPP"; UpdateQuickList: Yes!)
```

---

## PrefLocationOfTemplates

---

**Purpose**

PrefLocationOfTemplates specifies the location of template files.



---

**Syntax** PrefLocationOfTemplates (TemplateDir: *Character Expression*; TemplateSupplPath: *Character Expression*; File: *Character Expression*; SupplFile: *Character Expression*; DefExt: *Character Expression*; UpdateQuickList: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*TemplateDir* Character Expression: Specifies the default templates directory.

*TemplateSupplPath* Character Expression: Specifies the default templates supplemental directory.

*File* Character Expression: Specifies the default templates filename.

*SupplFile* Character Expression: Specifies the default supplemental filename.

*DefExt* Character Expression: Specifies the default filename extension.

*UpdateQuickList* Numeric Equivalent or Enumerated Type: Specifies whether to update QuickList with changes.

0 No!  
1 Yes!

---

**Route** File, Preferences, File, Templates, specify options.

---

**Example** To specify template names and the location of template files, the command is:

```
PrefLocationOfTemplates (TemplateDir: "C:\WPWIN60\TEMPLATE"; TemplateSupplPath:
"C:\WPWIN60\TEMPLATE\SUPP"; File: "TEMPLATE"; SupplFile: "TEMPSUPP"; DefExt: "EXT";
UpdateQuickList: Yes!)
```

---

---

## PrefMenu

**Purpose** PrefMenu specifies whether to display help prompts, shortcut keys next to menu items, and whether to display the four most recently opened documents at the bottom of the File menu.

---

**Syntax** PrefMenu (DisplayShortcuts: *Numeric Equivalent or Enumerated Type*, DisplayLastOpened: *Numeric Equivalent or Enumerated Type*, DisplayHelpPrompts: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*DisplayShortcuts* Numeric Equivalent or Enumerated Type

0 No!  
1 Yes!

---

*DisplayLastOpened* Numeric Equivalent or Enumerated Type

- 0 No!
- 1 Yes!

*DisplayHelpPrompts* Numeric Equivalent or Enumerated Type

- 0 No!
- 1 Yes!

---

**Route** File, Preferences, Environment, Menu, select options.

---

**Example** To set menu options, the command is:

PrefMenu (DisplayShortcuts: Yes!; DisplayLastOpened: Yes!; DisplayHelpPrompts: Yes!)

---

**See Also** ♦ PrefEnvSettingsDlg ♦ PrefSave

---

## PrefMergeCodes

---

**Purpose** PrefMergeCodes specifies whether to display merge codes.

---

**Syntax** PrefMergeCodes (Show: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

*Show* Numeric Equivalent or Enumerated Type: Specifies whether to hide merge codes, display merge codes, or display merge codes as markers.

- 0 Hide!
- 1 Show!
- 2 ShowAsBars!

---

**Route** File, Preferences, Display, Merge, select an option.

---

**Example** To display merge codes, the command is:

PrefMergeCodes (Show: Show!)

---

# PrefMetafileOptions

---

|                   |                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrefMetafileOptions specifies how to import graphics files.                                                                                                                                                                     |
| <b>Syntax</b>     | PrefMetafileOptions (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                               |
| <b>Parameters</b> |                                                                                                                                                                                                                                 |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to convert to WordPerfect (.WPG) format, retain both formats, or retain Metafile (.WMF) format.<br>0 ConvertToWPG!<br>1 RetainWindowsAndWPG!<br>2 RetainWindowsFormat! |
| <b>Route</b>      | File, Preferences, Import, select an option.                                                                                                                                                                                    |
| <b>Example</b>    | To convert graphics to .WPG format, the command is:<br><pre>PrefMetafileOptions (State: ConvertToWPG!)</pre>                                                                                                                    |

---

# PrefPowerBar

---

|                   |                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrefPowerBar specifies buttons to display on the Power Bar.                                                                                                                 |
| <b>Syntax</b>     | PrefPowerBar ({Button: <i>Numeric Expression</i> ; Position: <i>Numeric Expression</i> })                                                                                   |
| <b>Parameters</b> | Use the Button parameter followed by the Position parameter to define each button on the Power Bar. Enclose repeated parameters in braces {} separated by semicolons.       |
| <i>Button</i>     | Numeric Expression: Specifies a button number.                                                                                                                              |
| <i>Position</i>   | Numeric Expression: Specifies a button position on the Power Bar.                                                                                                           |
| <b>Route</b>      | File, Preferences, Power Bar, select Items, drag buttons to specify position.                                                                                               |
| <b>Example</b>    | To define a Power Bar with button 3 in position 4 and button 6 in position 7, the command is:<br><pre>PrefPowerBar ({Button: 3; Position: 4; Button: 6; Position: 7})</pre> |

---

## PrefPowerBarEdit

---

**Purpose** PrefPowerBarEdit opens and closes the Power Bar Preferences dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** PrefPowerBarEdit (Action: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to enter or exit the Power Bar Preferences dialog box.

- 0 Exit!
- 1 Enter!

**Example** To close the Power Bar Preferences dialog box, the command is:  
PrefPowerBarEdit (Action: Exit!)

---

## PrefPowerBarFontFaceList

---

**Purpose** PrefPowerBarFontFaceList specifies fonts to add to the Power Bar Font Face list.

**Syntax** PrefPowerBarFontFaceList ({FontName: *Character Expression*})

**Parameters**

*FontName* Character Expression: Specifies a font to add to the Power Bar Font Face list. Repeat this parameter for each font added. Enclose repeated parameters in braces {} separated by semicolons.

**Route** File, Preferences, Power Bar, Fonts, select options in the Fonts list box, or choose a font category in the Select pop-up list.

**Example** To add Helvetica and Arial to the Power Bar Font Face list, the command is:  
PrefPowerBarFontFaceList ({FontName: "Helvetica"; FontName: "Arial"})

---

# PrefPowerBarFontSizeList

---

|                   |                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrefPowerBarFontSizeList specifies a font size to add to the Power Bar Font Size list.                                                                                                              |
| <b>Syntax</b>     | PrefPowerBarFontSizeList ({Size: <i>Numeric Expression</i> })                                                                                                                                       |
| <b>Parameters</b> |                                                                                                                                                                                                     |
| <i>Size</i>       | Numeric Expression: Specifies a font size to add to the Power Bar Font Size list. Repeat this parameter for each font size added. Enclose repeated parameters in braces {} separated by semicolons. |
| <b>Route</b>      | File, Preferences, Power Bar, Fonts, select options in the Sizes list box, or click Select and choose a size category.                                                                              |
| <b>Example</b>    | To add font sizes to the Power Bar, the command is:<br><pre>PrefPowerBarFontSizeList ({Size: 8; Size: 10; Size: 12})</pre>                                                                          |

---

# PrefPrintCopies

---

|                       |                                                                                                                                           |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>        | PrefPrintCopies specifies the default number of copies to print and how to generate multiple copies.                                      |
| <b>Syntax</b>         | PrefPrintCopies (NumberOfCopies: <i>Numeric Expression</i> ; GeneratedBy: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )           |
| <b>Parameters</b>     |                                                                                                                                           |
| <i>NumberOfCopies</i> | Numeric Expression: Specifies the default number of copies to print.                                                                      |
| <i>GeneratedBy</i>    | Numeric Equivalent or Enumerated Type: Specifies if WordPerfect or the printer generates multiple copies.<br>0 WordPerfect!<br>1 Printer! |
| <b>Route</b>          | File, Preferences, Print, Copies, specify options.                                                                                        |
| <b>Example</b>        | To set print options, the command is:<br><pre>PrefPrintCopies (NumberOfCopies: 1; GeneratedBy: WordPerfect!)</pre>                        |
| <b>See Also</b>       | ♦ PrefPrintSettingsDlg ♦ PrefSave                                                                                                         |

---

# PrefPrintDoc

---

**Purpose** PrefPrintDoc specifies the default binding offset, graphics quality, and text quality for printed documents.

---

**Syntax** PrefPrintDoc (GraphicsQuality: *Numeric Equivalent* or *Enumerated Type*, TextQuality: *Numeric Equivalent* or *Enumerated Type*, PrintColor: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*GraphicsQuality* Numeric Equivalent or Enumerated Type: Specifies graphics print quality.

- 0 DoNotPrint!
- 1 Draft!
- 2 Medium!
- 3 High!

*TextQuality* Numeric Equivalent or Enumerated Type: Specifies text print quality.

- 0 DoNotPrint!
- 1 Draft!
- 2 Medium!
- 3 High!

*PrintColor* Numeric Equivalent or Enumerated Type: Specifies whether to print color.

- 0 Black!
  - 1 FullColor!
- 

**Route** File, Preferences, Print, Document Settings, select options.

---

**Example** To set print options, the command is:  
PrefPrintDoc (GraphicsQuality: High!; TextQuality: High!; PrintColor: Black!)

---

**See Also** ♦ PrefSave

---

---

# PrefPrintRedline

---

**Purpose** PrefPrintRedline specifies how and where redlined text is displayed.

---

**Syntax** PrefPrintRedline (Method: *Numeric Equivalent* or *Enumerated Type*, Character: *Character Expression*)

---

---

## Parameters

|                  |                                                                                                                                                                                                                                                                                                                                          |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Method</i>    | Numeric Equivalent or Enumerated Type: Specifies how redlined text is printed. You can accept the printer setting, or specify a column of characters down the left or alternating sides of redlined text.<br><ul style="list-style-type: none"><li>0 PrinterDependent!</li><li>1 Left!</li><li>2 Alternating!</li><li>3 Right!</li></ul> |
| <i>Character</i> | Character Expression: Specifies a character to display down the left or alternating sides of redlined text. Omit this parameter if the Method parameter is PrinterDependent!.                                                                                                                                                            |

---

**Route** Layout, Document, Redline Method, select a method, specify Redline Character if available, Use as Default.

---

**Example** To set print redline options, the command is:  
`PrefPrintRedline (Method: Left!; Character: "l")`

---

**See Also** ♦ PrefSave

---

## PrefPrintSettingsDlg

---

**Purpose** PrefPrintSettingsDlg displays the Print Preferences dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PrefPrintSettingsDlg ()

---

## PrefPrintSizeRatio

---

**Purpose** PrefPrintSizeRatio specifies size for each font attribute, as a percentage of the selected font. The value range for each font attribute is 1 to 999.

---

**Syntax** PrefPrintSizeRatio (Fine: *Numeric Expression*; Small: *Numeric Expression*; Large: *Numeric Expression*; VeryLarge: *Numeric Expression*; ExtraLarge: *Numeric Expression*; SuperSubscript: *Numeric Expression*)

---

## Parameters

|                       |                                                 |
|-----------------------|-------------------------------------------------|
| <i>Fine</i>           | Numeric Expression: Values range from 1 to 100. |
| <i>Small</i>          | Numeric Expression: Values range from 1 to 100. |
| <i>Large</i>          | Numeric Expression: Values range from 1 to 100. |
| <i>VeryLarge</i>      | Numeric Expression: Values range from 1 to 100. |
| <i>ExtraLarge</i>     | Numeric Expression: Values range from 1 to 100. |
| <i>SuperSubscript</i> | Numeric Expression: Values range from 1 to 100. |

---

## Route

File, Preferences, Print, Size Attribute Ratio, specify options.

---

## Example

To set print size ratio options, the command is:

PrefPrintSizeRatio (Fine: 25; Small: 50; Large: 125; VeryLarge: 150; ExtraLarge: 200; SuperSubscript: 15)

---

## See Also

♦ PrefSave

---

# PrefRevealCodes

---

## Purpose

PrefRevealCodes specifies the default Reveal Code settings.

---

## Syntax

PrefRevealCodes (FontFace: *Character Expression*; FontSize: *Numeric Expression*; WindowSize: *Numeric Expression*; ShowOnNew: *Numeric Equivalent* or *Enumerated Type*; ShowSpacesAsBullets: *Numeric Equivalent* or *Enumerated Type*; WrapAtWindow: *Numeric Equivalent* or *Enumerated Type*; ShowInDetail: *Numeric Equivalent* or *Enumerated Type*; Sculptured: *Numeric Equivalent* or *Enumerated Type*; ForeColorRed: *Numeric Expression*; ForeColorGreen: *Numeric Expression*; ForeColorBlue: *Numeric Expression*; BackColorRed: *Numeric Expression*; BackColorGreen: *Numeric Expression*; BackColorBlue: *Numeric Expression*; ShowHelpPrompts: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

|                   |                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <i>FontFace</i>   | Character Expression: Specifies a font face name.                                                                                     |
| <i>FontSize</i>   | Numeric Expression: Specifies a font height in points, where 72 points equals one inch.                                               |
| <i>WindowSize</i> | Numeric Expression: Specifies window size as a percentage of the screen.                                                              |
| <i>ShowOnNew</i>  | Numeric Equivalent or Enumerated Type: Specifies whether to turn on Reveal Codes in current and new documents.<br><br>0 No!<br>1 Yes! |



|                            |                                                                                                               |
|----------------------------|---------------------------------------------------------------------------------------------------------------|
| <i>ShowSpacesAsBullets</i> | Numeric Equivalent or Enumerated Type: Specifies whether to show spaces as bullets.<br>0 No!<br>1 Yes!        |
| <i>WrapAtWindow</i>        | Numeric Equivalent or Enumerated Type: Specifies whether to wrap lines at the window edge.<br>0 No!<br>1 Yes! |
| <i>ShowInDetail</i>        | Numeric Equivalent or Enumerated Type: Specifies whether to show codes in detail.<br>0 No!<br>1 Yes!          |
| <i>Sculptured</i>          | Numeric Equivalent or Enumerated Type: Specifies whether to display codes sculptured.<br>0 No!<br>1 Yes!      |
| <i>ForeColorRed</i>        | Numeric Expression: Specifies the amount of red (0-255) in a foreground color.                                |
| <i>ForeColorGreen</i>      | Numeric Expression: Specifies the amount of green (0-255) in a foreground color.                              |
| <i>ForeColorBlue</i>       | Numeric Expression: Specifies the amount of blue (0-255) in a foreground color.                               |
| <i>BackColorRed</i>        | Numeric Expression: Specifies the amount of red (0-255) in a background color.                                |
| <i>BackColorGreen</i>      | Numeric Expression: Specifies the amount of green (0-255) in a background color.                              |
| <i>BackColorBlue</i>       | Numeric Expression: Specifies the amount of blue (0-255) in a background color.                               |
| <i>ShowHelpPrompts</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to show help prompts.<br>0 No!<br>1 Yes!             |

---

**Route** File, Preferences, Display, Reveal Codes, select options.

---

**Example** To specify Reveal Code settings, the command is:

```
PrefRevealCodes (FontFace: "Arial"; FontSize: 15; WindowSize: 25; ShowOnNew: No!;
ShowSpacesAsBullets: Yes!; WrapAtWindow: Yes!; ShowInDetail: No!; Sculptured: Yes!; ForeColorRed: 0;
ForeColorGreen: 0; ForeColorBlue: 0; BackColorRed: 192; BackColorGreen: 192; BackColorBlue: 192;
ShowHelpPrompts: Yes!)
```

---

## PrefRevealCodesDlg

**Purpose** PrefRevealCodesDlg displays the Display Preferences dialog box with Reveal Codes selected.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** PrefRevealCodesDlg ()

---

**See Also** ♦ PrefRevealCodes

---

---

## PrefRuler

---

**Purpose** PrefRuler specifies default ruler bar settings.

---

**Syntax** PrefRuler (RulerOnNew: *Numeric Equivalent* or *Enumerated Type*; TabsToGrid: *Numeric Equivalent* or *Enumerated Type*; ShowRulerGuides: *Numeric Equivalent* or *Enumerated Type*; SculpturedRuler: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*RulerOnNew* Numeric Equivalent or Enumerated Type: Specifies whether to display the Ruler Bar when you open a new document.

0 No!  
1 Yes!

*TabsToGrid* Numeric Equivalent or Enumerated Type: Specifies whether tabs "snap to" invisible grid lines every 1/16th of an inch or 1/5th of a centimeter.

0 No!  
1 Yes!

*ShowRulerGuides* Numeric Equivalent or Enumerated Type: Specifies whether a Ruler Bar guide displays when tab stops move.

0 No!  
1 Yes!

*SculpturedRuler* Numeric Equivalent or Enumerated Type: Specifies whether to display a sculptured ruler.

0 No!  
1 Yes!

---

**Route** File, Preferences, Display, Ruler Bar, select options.

---

**Example** To set Ruler Bar options, the command is:

PrefRuler (RulerOnNew: Yes!; TabsToGrid: No!; ShowRulerGuides: No!; SculpturedRuler: Yes!)

---

**See Also** ♦ PrefEnvSettingsDlg ♦ PrefSave

---

---

# PrefSave

---

|                |                                                                                    |
|----------------|------------------------------------------------------------------------------------|
| <b>Purpose</b> | PrefSave saves preference command changes, to apply each time you run WordPerfect. |
| <b>Syntax</b>  | PrefSave ()                                                                        |
| <b>Route</b>   | Choose OK in any Preferences dialog box.                                           |

---

# PrefStatusBar

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrefStatusBar specifies default status bar settings.<br><br>IMPORTANT: This command must be preceded by and followed by PrefStatusBarEdit.                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Syntax</b>     | PrefStatusBar ({Box: <i>Numeric Equivalent or Enumerated Type</i> ; Position: <i>Numeric Expression</i> ; Width: <i>Numeric Expression</i> })                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Parameters</b> | Repeat the Box, Position, and Width parameters for each status bar item. Enclose repeated parameters in braces {} separated by semicolons.                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>Box</i>        | Numeric Equivalent or Enumerated Type: Specifies an item to display or hide on the status bar.<br><br><ul style="list-style-type: none"><li>0 Font!</li><li>1 Select!</li><li>2 InsertMode!</li><li>3 Date!</li><li>4 Time12!</li><li>5 GeneralStatus!</li><li>6 AlignChar!</li><li>7 Outline!</li><li>8 Zoom!</li><li>9 CapsLockState!</li><li>10 NumLockState!</li><li>11 ScrollLockState!</li><li>12 Printer!</li><li>13 Keyboard!</li><li>14 Page!</li><li>15 Line!</li><li>16 Position!</li><li>17 CombinedPosition!</li><li>18 Time24!</li></ul> |

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| <i>Position</i> | Numeric Expression: Specifies a position on the status bar. |
| <i>Width</i>    | Numeric Expression: Specifies an item's width.              |

---

**Route** File, Preferences, Status Bar, select status bar items.

---

**Example** To specify status bar settings, the command is:

```
PrefStatusBarEdit (State: On!)
PrefStatusBar (!Box: Date!; Position: -1; Width: -1; Box: Font! Position: -1; Width: -1!)
PrefSave ()
PrefStatusBarEdit (State: Off!)
```

---

## PrefStatusBarEdit

**Purpose** PrefStatusBarEdit must precede and follow the PrefStatusBar command. If no parameter is specified, the command acts as a toggle.

**Syntax** PrefStatusBarEdit (State: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*State* Numeric Equivalent or Enumerated Type

|   |      |
|---|------|
| 0 | Off! |
| 1 | On!  |

---

**Route** File, Preferences, Status Bar.

**Example** To edit the status bar, the command is:

```
PrefStatusBarEdit (State: On!)
```

---

## PrefStatusBarOptions

**Purpose** PrefStatusBarOptions specifies status bar settings.

**Syntax** PrefStatusBarOptions (FontFace: *Character Expression*; FontSize: *Numeric Expression*; Bold: *Numeric Equivalent* or *Enumerated Type*; BoxStyle: *Numeric Equivalent* or *Enumerated Type*; EvenSpaced: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <i>FontFace</i>   | Character Expression: Specifies the name of a font face.                                                              |
| <i>FontSize</i>   | Numeric Expression: Specifies a font height in points, where 72 points equals one inch.                               |
| <i>Bold</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to bold the status bar font.<br>0 No!<br>1 Yes!              |
| <i>BoxStyle</i>   | Numeric Equivalent or Enumerated Type: Specifies how the status bar is displayed.<br>0 Flat!<br>1 Inset!<br>2 Raised! |
| <i>EvenSpaced</i> | Numeric Equivalent or Enumerated Type: Specifies whether to space items evenly.<br>0 No!<br>1 Yes!                    |

---

**Route** File, Preferences, Status Bar, Options, select options.

---

**Example** To specify status bar options, the command is:  
`PrefStatusBarOptions (FontFace: "Courier"; FontSize: 12p; Bold: Yes!; BoxStyle: Raised!; EvenSpaced: No!)`

---

**See Also** ♦ `PrefStatusBarEdit`

---

## PrefStatusBarOptionsDlg

---

**Purpose** `PrefStatusBarOptionsDlg` displays the Status Bar Options dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** `PrefStatusBarOptionsDlg ()`

---

# PrefToA

|                    |                                                                                                                                                      |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | PrefToA specifies a table of authorities format.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>      | PrefToA (Underlining: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Runs: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )              |
| <b>Parameters</b>  |                                                                                                                                                      |
| <i>Underlining</i> | Numeric Equivalent or Enumerated Type: Specifies whether to allow underlining in the full form.<br>0 NotAllowed!<br>1 Allowed!                       |
| <i>Runs</i>        | Numeric Equivalent or Enumerated Type<br>0 NotAllowed!<br>1 Allowed!                                                                                 |
| <b>Example</b>     | To set table of authorities options, the command is:<br><br>PrefToA (Underlining: Allowed! Runs: NotAllowed!)                                        |

---

# PrefUnitsOfMeasure

|                     |                                                                                                                                                                                             |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>      | PrefUnitsOfMeasure specifies units of measure settings.                                                                                                                                     |
| <b>Syntax</b>       | PrefUnitsOfMeasure (DisplayUnits: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , StatusBarUnits: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                               |
| <b>Parameters</b>   |                                                                                                                                                                                             |
| <i>DisplayUnits</i> | Numeric Equivalent or Enumerated Type: Specifies default units of measure and how to display them.<br>0 InchesQuote!<br>1 Inches!<br>2 Centimeters!<br>3 Millimeters!<br>4 Point!<br>5 WPU! |

## StatusBarUnits

Numeric Equivalent or Enumerated Type: Specifies how units of measure are displayed on the status bar and ruler.

- 0 InchesQuote!
- 1 Inches!
- 2 Centimeters!
- 3 Millimeters!
- 4 Point!
- 5 WPU!

---

**Route** File, Preferences, Display, Document, Measurement, select options.

---

**Example** To specify units of measure, the command is:

PrefUnitsOfMeasure (DisplayUnits: InchesQuote!; StatusBarUnits: InchesQuote!)

---

## PrefZoom

---

**Purpose** PrefZoom enlarges or reduces the size of text and graphics.

---

**Syntax** PrefZoom (Type: *Numeric Equivalent* or *Enumerated Type*; Percent: *Numeric Expression*)

---

### Parameters

*Type* Numeric Equivalent or Enumerated Type: Specifies a type of zoom.

- 0 FixedPercent!
- 1 MarginWidth!
- 2 PageWidth!
- 3 FullPage!

*Percent* Numeric Expression: Specifies a percentage of normal, which is 100 percent.

---

**Route** File, Preferences, Display, View/Zoom, select a Default Zoom.

---

**Example** To set zoom options, the command is:

PrefZoom (Type: FixedPercent!; Percent: 150)

---

**See Also** ♦ DraftZoom100

---

# PrintCopies

---

**Purpose** PrintCopies specifies the number of copies to print.

---

**Syntax** PrintCopies (NumberOfCopies: *Numeric Expression*)

---

## Parameters

*NumberOfCopies* Numeric Expression: Specifies the number of copies to print.

---

**Route** File, Print, Number of Copies, enter a number.

---

**Example** To print two copies of a document, the command is:

PrintCopies (NumberOfCopies: 2)

---

**See Also** ♦ PrintCopiesBy

---

# PrintCopiesBy

---

**Purpose** PrintCopiesBy determines whether WordPerfect or a printer generates multiple copies.

---

**Syntax** PrintCopiesBy (CopiesBy: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*CopiesBy* Numeric Equivalent or Enumerated Type: Specifies whether WordPerfect or a printer generates multiple copies.

- 0 WordPerfect!
- 1 Printer!
- 2 Network!

---

**Route** File, Print, Generated By, select an option.

---

**Example** To specify WordPerfect to generate multiple copies, the command is:

PrintCopiesBy (CopiesBy: WordPerfect!)

---

**See Also** ♦ PrintCopies



---

# PrintDlg

---

**Purpose** PrintDlg displays the Print dialog box.

---

**Syntax** PrintDlg ()

---

**Route** File, Print.

---

---

# PrintDocOnDisk

---

**Purpose** PrintDocOnDisk prints a document saved to disk.

---

**Syntax** PrintDocOnDisk (Filename: *Character Expression*; PageRange: *Character Expression*; SecondaryRange: *Character Expression*; ChapterRange: *Character Expression*; VolumeRange: *Character Expression*; OddEvenPages: *Numeric Equivalent* or *Enumerated Type*; PrintSummary: *Numeric Equivalent* or *Enumerated Type*; PrintBooklet: *Numeric Equivalent* or *Enumerated Type*; Direction: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*Filename* Character Expression: Specifies the name of a file to print.

*PageRange* Character Expression: Specifies a range of pages to print.

*SecondaryRange* Character Expression: Specifies the name of an output file.

*ChapterRange* Character Expression: Specifies a range of chapters to print.

*VolumeRange* Character Expression: Specifies a volume range to print.

*OddEvenPages* Numeric Equivalent or Enumerated Type: Determines pages to print.

- 0 PrintAll!
- 2 PrintOdd!
- 4 PrintEven!

*PrintSummary* Numeric Equivalent or Enumerated Type: Specifies whether to print an existing document summary.

- 0 NoDocSummary!
- 1 DocSummary!

*PrintBooklet* Numeric Equivalent or Enumerated Type: Specifies whether to print the document as a booklet.

- 0 NoBooklet!
- 1 Booklet!

|                  |                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <i>Direction</i> | Numeric Equivalent or Enumerated Type: Specifies whether to print pages in ascending or descending order.<br>0 Forward!<br>1 Backward! |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------|

|              |                                                                                            |
|--------------|--------------------------------------------------------------------------------------------|
| <b>Route</b> | File, Print, Document on Disk, Print, specify a filename, specify a range of pages, Print. |
|--------------|--------------------------------------------------------------------------------------------|

|                |                                                                                              |
|----------------|----------------------------------------------------------------------------------------------|
| <b>Example</b> | To print all pages of a document on disk named TEST with a document summary, the command is: |
|----------------|----------------------------------------------------------------------------------------------|

PrintDocOnDisk (Filename: "TEST"; PageRange: "1-10"; SecondaryRange: ""; ChapterRange: ""; VolumeRange: ""; OddEvenPages: PrintAll!; PrintSummary: DocSummary!; PrintBooklet: NoBooklet!; Direction: Forward!)

---

## PrinterCommand

|                |                                                                                |
|----------------|--------------------------------------------------------------------------------|
| <b>Purpose</b> | PrinterCommand inserts a printer command [Ptr Cmnd] into the current document. |
|----------------|--------------------------------------------------------------------------------|

|               |                                                                                                                    |
|---------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | PrinterCommand (Flag: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Command: <i>Character Expression</i> ) |
|---------------|--------------------------------------------------------------------------------------------------------------------|

### Parameters

|             |                                                                                                                                                                    |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Flag</i> | Numeric Equivalent or Enumerated Type: Specifies whether the Command parameter contains a printer command or the name of a file with one or more printer commands. |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- 0 File!
- 1 Command!

|                |                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <i>Command</i> | Character Expression or Enumerated Type: Specifies a printer command or the name of a file containing one or more printer commands. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|

|              |                                                                                                                                              |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Layout, Typesetting, Printer Command, select Command and enter a printer command, or select Printer Command Filename and specify a filename. |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|

|                |                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------|
| <b>Example</b> | To insert printer commands into the current document using a filename PTRCMD, the command is: |
|----------------|-----------------------------------------------------------------------------------------------|

PrinterCommand (Flag: File!; Command: "C:\WP60\PTRCMD")

---

## PrinterCommandDlg

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <b>Purpose</b>  | PrinterCommandDlg displays the Printer Command dialog box. |
| <b>Syntax</b>   | PrinterCommandDlg ()                                       |
| <b>Route</b>    | Layout, Typesetting, Printer Command.                      |
| <b>See Also</b> | ♦ PrinterCommand                                           |

---

## PrinterControlDlg

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <b>Purpose</b> | PrinterControlDlg displays the WP Print Process dialog box. |
| <b>Syntax</b>  | PrinterControlDlg ()                                        |
| <b>Route</b>   | File, Print, Control.                                       |

---

## PrinterInitialize

|                |                                                        |
|----------------|--------------------------------------------------------|
| <b>Purpose</b> | PrinterInitialize downloads soft fonts to the printer. |
| <b>Syntax</b>  | PrinterInitialize ()                                   |
| <b>Route</b>   | File, Print, Initialize.                               |

---

## PrinterPause

|                |                                                      |
|----------------|------------------------------------------------------|
| <b>Purpose</b> | PrinterPause pauses the current print job.           |
| <b>Syntax</b>  | PrinterPause ()                                      |
| <b>Route</b>   | Layout, Typesetting, Printer Command, Pause Printer. |

---

## PrinterSelect

---

|                    |                                                                                                                                                                                                   |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | PrinterSelect specifies a printer type, name, and port.                                                                                                                                           |
| <b>Syntax</b>      | PrinterSelect (PrinterType: <i>Numeric Equivalent or Enumerated Type</i> , PrinterName: <i>Character Expression</i> ; PrinterPort: <i>Character Expression</i> )                                  |
| <b>Parameters</b>  |                                                                                                                                                                                                   |
| <i>PrinterType</i> | Numeric Equivalent or Enumerated Type: Specifies whether to send to a WordPerfect or Windows print driver.<br>0 WPrinter!<br>1 WindowsPrint!                                                      |
| <i>PrinterName</i> | Character Expression: Specifies a printer name.                                                                                                                                                   |
| <i>PrinterPort</i> | Character Expression: Specifies a printer port.                                                                                                                                                   |
| <b>Route</b>       | File, Select Printer, select a printer, Select.                                                                                                                                                   |
| <b>Example</b>     | To select the Apple LaserWriter II NTX windows printer driver, the command is:<br><br>PrinterSelect (PrinterType: WindowsPrinter!; PrinterName: "Apple LaserWriter II NTX"; PrinterPort: "LPT1:") |

---

## PrinterSelectDlg

---

|                |                                                          |
|----------------|----------------------------------------------------------|
| <b>Purpose</b> | PrinterSelectDlg displays the Select Printer dialog box. |
| <b>Syntax</b>  | PrinterSelectDlg ()                                      |
| <b>Route</b>   | File, Select Printer.                                    |

---

## PrintFullDoc

---

|                |                                           |
|----------------|-------------------------------------------|
| <b>Purpose</b> | PrintFullDoc prints the current document. |
| <b>Syntax</b>  | PrintFullDoc ()                           |
| <b>Route</b>   | File, Print, Full Document, Print.        |

---

---

# PrintGraphicsQuality

---

|                   |                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrintGraphicsQuality specifies graphics print quality.                                                                     |
| <b>Syntax</b>     | PrintGraphicsQuality (Quality: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                       |
| <b>Parameters</b> |                                                                                                                            |
| <i>Quality</i>    | Numeric Equivalent or Enumerated Type: Specifies graphics resolution.<br>0 DoNotPrint!<br>1 Draft!<br>2 Medium!<br>3 High! |
| <b>Route</b>      | File, Print, Print Quality, select a print quality.                                                                        |
| <b>Example</b>    | To print graphics with high resolution, the command is:<br>PrintGraphicsQuality (Quality: High!)                           |
| <b>See Also</b>   | ♦ PrintTextQuality                                                                                                         |

---

# PrintInkColor

---

|                   |                                                                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | PrintInkColor specifies whether to print black, full color, or solid color (non-dithered).                                                                                                      |
| <b>Syntax</b>     | PrintInkColor (ColorType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                 |
| <b>Parameters</b> |                                                                                                                                                                                                 |
| <i>ColorType</i>  | Numeric Equivalent or Enumerated Type: Specifies a color to print. SolidColor! is printer-specific and may not be available on all color printers.<br>0 Black!<br>1 FullColor!<br>2 SolidColor! |
| <b>Route</b>      | File, Print, Print Color, select a color.                                                                                                                                                       |
| <b>Example</b>    | To print solid color, the command is:<br>PrintInkColor (ColorType: SolidColor!)                                                                                                                 |

---

# PrintMultiplePages

---

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | PrintMultiplePages prints specified pages.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Syntax</b>  | PrintMultiplePages (PageRange: <i>Character Expression</i> ; SecondaryPageRange: <i>Character Expression</i> ; ChapterRange: <i>Character Expression</i> ; VolumeRange: <i>Character Expression</i> ; OddEvenPages: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; PrintSummary: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; PrintBooklet: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Direction: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

## Parameters

|                           |                                                                                                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <i>PageRange</i>          | Character Expression: Specifies a range of pages to print.                                                                             |
| <i>SecondaryPageRange</i> | Character Expression: Specifies the name of an output file.                                                                            |
| <i>ChapterRange</i>       | Character Expression: Specifies a range of chapters to print.                                                                          |
| <i>VolumeRange</i>        | Character Expression: Specifies a range of volumes to print.                                                                           |
| <i>OddEvenPages</i>       | Numeric Equivalent or Enumerated Type: Specifies pages to print.<br>0 PrintAll!<br>2 PrintOdd!<br>4 PrintEven!                         |
| <i>PrintSummary</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to print an existing document summary.<br>0 NoDocSummary!<br>1 DocSummary!    |
| <i>PrintBooklet</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to print the document as a booklet.<br>0 NoBooklet!<br>1 Booklet!             |
| <i>Direction</i>          | Numeric Equivalent or Enumerated Type: Specifies whether to print pages in ascending or descending order.<br>0 Forward!<br>1 Backward! |

---

**Route** File, Print, Multiple Pages, Print, specify a range, Print.

---

**Example** To set print options, the command is:

```
PrintMultiplePages (PageRange: "1-10"; SecondaryPageRange: "PRTFILE"; ChapterRange: "1-2";
VolumeRange: "1"; OddEvenPages: PrintAll!; PrintSummary: NoDocSummary!; PrintBooklet: NoBooklet!;
Direction: Forward!)
```

---

## PrintPage

---

**Purpose** PrintPage prints the current page.

---

**Syntax** PrintPage ()

---

**Route** File, Print, Current Page, Print.

---

---

## PrintSelection

---

**Purpose** PrintSelection prints selected text.  
IMPORTANT: If no text is selected, this command ends the macro or redirects it to the LABEL specified by ONERROR.

---

**Syntax** PrintSelection ()

---

**Route** With text selected, File, Print, Print.

---

---

## PrintTextQuality

---

**Purpose** PrintTextQuality specifies text print quality.

---

**Syntax** PrintTextQuality (Quality: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Quality* Numeric Equivalent or Enumerated Type: Specifies printed text resolution.

- 0 DoNotPrint!
- 1 Draft!
- 2 Medium!
- 3 High!

---

**Route** File, Print, Print Quality, select an option.

---

**Example** To print the current document text with high resolution, the command is:

PrintTextQuality (Quality: High!)

---

**See Also**

♦ PrintGraphicsQuality

---

## QuickFinderDlg

---

**Purpose**

QuickFinderDlg displays the QuickFinder dialog box.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax**

QuickFinderDlg ()

---

## QuickFormat

---

**Purpose**

QuickFormat formats text based on fonts and attributes, or paragraph styles, current at the insertion point.

**Syntax**

QuickFormat (Format: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Format*

Numeric Equivalent or Enumerated Type: Specifies whether to format text based on current fonts and attributes, paragraph styles, or both.

- 0 Off!
  - 1 Attributes!
  - 2 Styles!
  - 3 StylesAndAttributes!
- 

**Route**

Layout, QuickFormat, select an option.

---

**Example**

To format text based on current paragraph styles, the command is:

QuickFormat (Format: Styles!)

---

## QuickFormatDlg

---

**Purpose**

QuickFormatDlg displays the QuickFormat dialog box.

---



---

|               |                      |
|---------------|----------------------|
| <b>Syntax</b> | QuickFormatDlg ()    |
| <b>Route</b>  | Layout, QuickFormat. |

---

## QuickFormatToggle

---

|                |                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | QuickFormatToggle displays the QuickFormat dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | QuickFormatToggle ()                                                                                                                                       |

---

## QuickmarkFind

---

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| <b>Purpose</b> | QuickmarkFind moves the insertion point to the first character after a QuickMark. |
| <b>Syntax</b>  | QuickmarkFind ()                                                                  |
| <b>Route</b>   | Insert, Bookmark, Find QuickMark.                                                 |

---

## QuickmarkSet

---

|                |                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | QuickmarkSet places a unique bookmark in a document.<br>IMPORTANT: If Set QuickMark in Document on Save is selected, any QuickMark added by the macro moves to the current insertion point when the document is saved. |
| <b>Syntax</b>  | QuickmarkSet ()                                                                                                                                                                                                        |
| <b>Route</b>   | Insert, Bookmark, Set QuickMark.                                                                                                                                                                                       |

---

---

## Redisplay

|                |                                                                        |
|----------------|------------------------------------------------------------------------|
| <b>Purpose</b> | Redisplay rewrites screen contents without moving the insertion point. |
| <b>Syntax</b>  | Redisplay ()                                                           |
| <b>Route</b>   | Press Ctrl+F3.                                                         |

---

## RedlineCharacter

|                   |                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | RedlineCharacter specifies a character to redline text.                                                                                                 |
| <b>Syntax</b>     | RedlineCharacter (Character: <i>Character Expression</i> )                                                                                              |
| <b>Parameters</b> |                                                                                                                                                         |
| <i>Character</i>  | Character Expression: Specifies a character.                                                                                                            |
| <b>Route</b>      | Layout, Document, Redline Method, Mark Left Margin or Mark Alternating Margins or Mark Right Margin, Redline Character, type character, Use as Default. |
| <b>Example</b>    | To select   as the redline character, the command is:<br><pre>RedlineCharacter (Character: " ")</pre>                                                   |
| <b>See Also</b>   | ♦ RedlineMethod ♦ ?RedlineCharacter                                                                                                                     |

---

## RedlineMethod

|                      |                                                                                                                                                                                                               |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>       | RedlineMethod specifies a redline method.                                                                                                                                                                     |
| <b>Syntax</b>        | RedlineMethod (RedlineMethod: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                           |
| <b>Parameters</b>    |                                                                                                                                                                                                               |
| <i>RedlineMethod</i> | Numeric Equivalent or Enumerated Type: Specifies a redline method.<br><ul style="list-style-type: none"><li>0 PrinterDependent!</li><li>1 LeftMargin!</li><li>2 Alternating!</li><li>3 RightMargin!</li></ul> |

---

|                 |                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Document, Redline Method, select a method.                                                                         |
| <b>Example</b>  | To specify redline markings on the right side, the command is:<br><code>RedlineMethod (RedlineMethod: RightMargin!)</code> |
| <b>See Also</b> | ♦ RedlineCharacter ♦ ?RedlineMethod                                                                                        |

---

## RepeatDlg

---

|                 |                                           |
|-----------------|-------------------------------------------|
| <b>Purpose</b>  | RepeatDlg displays the Repeat dialog box. |
| <b>Syntax</b>   | RepeatDlg ()                              |
| <b>Route</b>    | Edit, Repeat.                             |
| <b>See Also</b> | ♦ RepeatValue                             |

---

## RepeatSet

---

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| <b>Purpose</b>    | RepeatSet specifies the Repeat default number.                                       |
| <b>Syntax</b>     | RepeatSet (Value: <i>Numeric Expression</i> )                                        |
| <b>Parameters</b> |                                                                                      |
| <i>Value</i>      | Numeric Expression: Specifies the number of times to repeat an action.               |
| <b>Route</b>      | Edit, Repeat, enter a number, Use as Default.                                        |
| <b>Example</b>    | To set the Repeat default at 5, the command is:<br><code>RepeatSet (Value: 5)</code> |
| <b>See Also</b>   | ♦ RepeatValue                                                                        |

---

---

## RepeatValue

---

**Purpose** RepeatValue specifies the number of times to repeat an action.

---

**Syntax** RepeatValue (Value: *Numeric Expression*)

---

### Parameters

*Value* Numeric Expression: RepeatDlg

---

**Route** Edit, Repeat, specify a number.

---

**Example** To repeat an action 50 times, the command is:

RepeatValue (Value: 50)

---

## ReplaceAll

---

**Purpose** ReplaceAll replaces all occurrences of a matched word, code, or phrase.

IMPORTANT: This command must be preceded by other commands which define a search, such as SearchString, ReplaceString, and SearchNext.

---

**Syntax** ReplaceAll (SearchMode: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*SearchMode* Numeric Equivalent or Enumerated Type: Specifies the scope of a search. An extended search includes substructures such as headers or footnotes.

0 Regular!  
1 Extended!

---

**Route** Edit, Replace, enter search text or codes in Find, enter replacement text or codes in Replace With, Replace All.

---

**Example** To replace all occurrences of the word "do" with "do not", starting from the insertion point and excluding substructures, the command sequence is:

SearchString (StrgToLookFor: "do")  
ReplaceString (RplcStrg: "do not")  
ReplaceAll (SearchMode: Regular!)

---

**See Also** • SearchString • ReplaceString • SearchNext

---

# ReplaceBackward

---

**Purpose**

ReplaceBackward replaces specified text or codes from the current insertion point to the beginning of the document.

IMPORTANT: This command must be preceded by other commands which define a search, such as SearchString and ReplaceString.

**Syntax**

ReplaceBackward (SearchMode: *Numeric Equivalent* or *Enumerated Type*)

**Parameters***SearchMode*

Numeric Equivalent or Enumerated Type: Specifies the scope of a backward search. An extended search includes substructures such as headers or footnotes.

- 0 Regular!
- 1 Extended!

**Route**

Edit, Replace, enter search text or codes in Find, enter replacement text or codes in Replace With, Direction, Backward, Replace All.

**Example**

To replace "No" with "Yes" from the insertion point backward and including substructures, the command sequence is:

```
SearchString (StrgToLookFor: "No")
ReplaceString (RplcStrg: "Yes")
ReplaceBackward (SearchMode: Extended!)
```

**See Also**

- ♦ ReplaceCurrent ♦ ReplaceForward ♦ ReplaceString ♦ SearchPrevious
- ♦ SearchString

---

# ReplaceCurrent

---

**Purpose**

ReplaceCurrent replaces a matched word, code, or phrase.

IMPORTANT: This command must be preceded by other commands which define a search, such as SearchString, SearchReplace, and SearchNext.

**Syntax**

ReplaceCurrent ()

**Route**

Edit, Replace, enter search text or codes in Find, enter replacement text or codes in Replace With, Find, Replace.

---

**See Also**

♦ ReplaceString ♦ SearchNext ♦ SearchString ♦ ReplaceWithAttributes

---

## ReplaceDlg

**Purpose**

ReplaceDlg displays the Find and Replace Text dialog box.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax**

ReplaceDlg()

---

## ReplaceForward

**Purpose**

ReplaceForward replaces specified text or codes from the insertion point to the end of the document.

IMPORTANT: This command must be preceded by other commands which define a search, such as SearchString or ReplaceString.

**Syntax**

ReplaceForward (SearchMode: *Numeric Equivalent* or *Enumerated Type*)

**Parameters***SearchMode*

Numeric Equivalent or Enumerated Type: Specifies the scope of a forward search and replace. An extended search includes substructures such as headers or footnotes.

- 0 Regular!
- 1 Extended!

**Route**

Edit, Replace, enter search text or codes in Find, enter replacement text or codes in Replace With, Replace All. To deselect a backward search: Edit, Replace, enter text or codes in Find, enter text or codes in Replace With, Direction, Backward, Replace All.

**Example**

To replace "No" with "Yes" from the insertion point forward including substructures, the command sequence is:

```
SearchString (StrgToLookFor: "No")
ReplaceString (RplcStrg: "Yes")
ReplaceForward (SearchMode: Extended!)
```

---

**See Also**

♦ ReplaceBackward ♦ ReplaceString ♦ SearchString

---

# ReplaceString

|                   |                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ReplaceString specifies replacement text or codes.                                                                          |
| <b>Syntax</b>     | ReplaceString (RplcStrg: <i>Character Expression</i> )                                                                      |
| <b>Parameters</b> |                                                                                                                             |
| <i>RplcStrg</i>   | Character Expression: Specifies replacement text or codes.                                                                  |
| <b>Route</b>      | Edit, Replace, enter search text or codes in Find, enter replacement text or codes in Replace With, Replace or Replace All. |
| <b>Example</b>    | To specify replacement text as Jonathan, the command is:<br>ReplaceString (RplcStrg: "Jonathan")                            |
| <b>See Also</b>   | ♦ SearchString                                                                                                              |

---

# ReplaceWithAttributes

|                   |                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ReplaceWithAttributes specifies attributes for replacement text.                                                                                                                                                                                           |
| <b>Syntax</b>     | ReplaceWithAttributes (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Attribute: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                       |
| <b>Parameters</b> |                                                                                                                                                                                                                                                            |
| <i>State</i>      | Numeric Equivalent or Enumerated Type<br>0 No!<br>1 Yes!                                                                                                                                                                                                   |
| <i>Attribute</i>  | Numeric Equivalent or Enumerated Type: Enclose multiple attributes in braces ({}), separated by semicolons.<br>0 ExtraLarge!<br>1 VeryLarge!<br>2 Large!<br>3 Small!<br>4 Fine!<br>5 Superscript!<br>6 Subscript!<br>7 Outline!<br>8 Italics!<br>9 Shadow! |

- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!
- 16 Every!

---

**Route** Edit, Replace, enter search text in Find, enter replacement text in Replace With, choose Font from Replace, select attributes, OK, Replace or Replace All.

---

**Example** To underline replacement text, the command is:

ReplaceWithAttributes (State: Yes!; Attribute: Underline!)

---

**See Also** ♦ ReplaceCurrent ♦ ReplaceString ♦ SearchNext ♦ SearchString

---

## ReplaceWithCase

---

**Purpose** ReplaceWithCase specifies whether text replacement is case-sensitive.

---

**Syntax** ReplaceWithCase (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type

- 0 No!
  - 1 Yes!
- 

**Route** Edit, Replace, enter search text in Find, enter replacement text in Replace With, choose Case from Replace, Replace or Replace All.

---

**Example** To replace a word or phrase with uppercase or mixed case text, depending on the case of the search text, the command is:

ReplaceWithCase (State: Yes!)

---



# ReplaceWithFont

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ReplaceWithFont specifies the font and attributes of replacement text.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Syntax</b>     | ReplaceWithFont (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Name: <i>Character Expression</i> ; Family: <i>Numeric Expression</i> ; Attributes: <i>Numeric Expression</i> ; Weight: <i>Numeric Expression</i> ; Width: <i>Numeric Expression</i> ; Source: <i>Numeric Expression</i> ; Type: <i>Numeric Expression</i> ; CharacterSet: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>State</i>      | Numeric Equivalent or Enumerated Type<br>0 No!<br>1 Yes!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <i>Name</i>       | Character Expression: Specifies a font name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>Family</i>     | Numeric Equivalent or Enumerated Type<br>256 FamilyITCBenguiat!<br>2304 FamilyTimesRoman!<br>4352 FamilyITCGaramond!<br>6400 FamilyCenturySchoolbook!<br>8448 FamilyBodon!<br>10496 FamilyITCLubalinGraph!<br>12544 FamilyAlbertus!<br>14592 FamilyCourier!<br>16640 FamilyHelvetica!<br>18688 FamilyITCRonda!<br>20736 FamilyMicrostyle!<br>22784 FamilyOptima!<br>24832 FamilyLetterGothic!<br>26880 FamilyBrush!<br>28928 FamilyParkAvenue!<br>30976 FamilySignetRoundhand!<br>33024 FamilyTekton!<br>35072 FamilyUncial!<br>37120 FamilyOldEnglish!<br>39168 FamilyITCZapfDingbats!<br>41216 FamilyBroadway!<br>43264 FamilyComputer!<br>45312 FamilyPTBarnum!<br>47360 FamilyPonderosa!<br>49408 FamilyMadrone!<br>51456 FamilyHobo!<br>53504 FamilyRevue!<br>55552 FamilyAestheticOrnamented! |

## *Attributes*

### Numeric Equivalent or Enumerated Type

|   |                     |
|---|---------------------|
| 0 | FontMatchNormal!    |
| 1 | FontMatchItalic!    |
| 2 | FontMatchOutline!   |
| 4 | FontMatchShadow!    |
| 8 | FontMatchSmallCaps! |

## *Weight*

### Numeric Equivalent or Enumerated Type

|     |                   |
|-----|-------------------|
| 0   | WeightUnknown!    |
| 1   | WeightUltraThin!  |
| 8   | WeightUltraLight! |
| 16  | WeightExtraThin!  |
| 24  | WeightThin!       |
| 32  | WeightExtraLight! |
| 48  | WeightLight!      |
| 64  | WeightDemiLight!  |
| 80  | WeightSemiLight!  |
| 88  | WeightBook!       |
| 96  | WeightRegular!    |
| 104 | WeightRoman!      |
| 112 | WeightMedium!     |
| 128 | WeightDemiBold!   |
| 136 | WeightSemiBold!   |
| 144 | WeightBold!       |
| 160 | WeightExtraBold!  |
| 176 | WeightHeavy!      |
| 192 | WeightExtraHeavy! |
| 208 | WeightBlack!      |
| 224 | WeightExtraBlack! |
| 240 | WeightUltraHeavy! |
| 248 | WeightUltraBlack! |
| 254 | WeightMaximum!    |

## *Width*

### Numeric Equivalent or Enumerated Type

|     |                       |
|-----|-----------------------|
| 0   | WidthUnknown!         |
| 16  | WidthUltraCondensed!  |
| 32  | WidthUltraCompressed! |
| 48  | WidthCompressed!      |
| 64  | WidthExtraCondensed!  |
| 80  | WidthCondensed!       |
| 96  | WidthSemiCondensed!   |
| 112 | WidthNormal!          |
| 128 | WidthSemiExpanded!    |
| 144 | WidthExpanded!        |
| 160 | WidthExtraExpanded!   |
| 176 | WidthUltraExpanded!   |
| 192 | WidthDoubleWide!      |

|                     |                                       |                              |
|---------------------|---------------------------------------|------------------------------|
|                     | 224                                   | WidthTripleWide!             |
| <i>Source</i>       | Numeric Equivalent or Enumerated Type |                              |
|                     | 16                                    | PRSGlobal!                   |
|                     | 20                                    | DRSGlobal!                   |
| <i>Type</i>         | Numeric Equivalent or Enumerated Type |                              |
|                     | 134                                   | Speedo!                      |
|                     | 135                                   | Intellifont!                 |
|                     | 138                                   | Type1!                       |
|                     | 139                                   | TrueType!                    |
| <i>CharacterSet</i> | Numeric Equivalent or Enumerated Type |                              |
|                     | 0                                     | FontMatchASCII!              |
|                     | 1                                     | FontMatchMultinational1!     |
|                     | 2                                     | FontMatchMultinational2!     |
|                     | 3                                     | FontMatchBoxDrawing!         |
|                     | 4                                     | FontMatchTypographicSymbols! |
|                     | 5                                     | FontMatchIconicSymbols!      |
|                     | 6                                     | FontMatchMath!               |
|                     | 7                                     | FontMatchMathExtension!      |
|                     | 8                                     | FontMatchGreek!              |
|                     | 9                                     | FontMatchHebrew!             |
|                     | 10                                    | FontMatchCyrillic!           |
|                     | 11                                    | FontMatchJapaneseKana!       |
|                     | 12                                    | FontMatchUserDefined!        |
|                     | 13                                    | FontMatchArabic!             |
|                     | 14                                    | FontMatchArabicScript!       |

---

**Route** Edit, Replace, enter search text in Find, enter replacement text in Replace With, choose Font from Replace, specify a font, OK, Replace or Replace All.

---

**Example** To specify the font and attributes for replacement text, the command is:

ReplaceWithFont (State: Yes!; Name: "Courier New Bold"; Family: FamilyCourier!; Attributes: FontMatchNormal!; Weight: WeightExtraBold!; Width: WidthUnknown!; Source: DRSFile!; Type: TrueType!; CharacterSet: FontMatchASCII!)

---

## ReplaceWithFontSize

---

**Purpose** ReplaceWithFontSize specifies the font size of replacement text.

|                   |                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | ReplaceWithFontSize (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , FontSize: <i>Numeric Expression</i> )                       |
| <b>Parameters</b> |                                                                                                                                               |
| <i>State</i>      | Numeric Equivalent or Enumerated Type<br>0 No!<br>1 Yes!                                                                                      |
| <i>FontSize</i>   | Numeric Expression: Specifies a font height in points, where 72 points equals one inch.                                                       |
| <b>Route</b>      | Edit, Replace, enter search text in Find, enter replacement text in Replace With, choose Font from Replace, Point Size, specify a point size. |
| <b>Example</b>    | To specify 12 point replacement text, the command is:<br>ReplaceWithFontSize (State: Yes!; FontSize: 12)                                      |

## ReselectLastSelection

|                |                                                       |
|----------------|-------------------------------------------------------|
| <b>Purpose</b> | ReselectLastSelection selects the last selected text. |
| <b>Syntax</b>  | ReselectLastSelection ()                              |
| <b>Route</b>   | Edit, Go To, Reselect Last Selection.                 |

## ResetKeyboardMenu

|                |                                                                                                                                                                          |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ResetKeyboardMenu resets the keyboard and menus to default settings.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | ResetKeyboardMenu ()                                                                                                                                                     |

---

# RevealCodes

**Purpose** RevealCodes specifies whether Reveal Codes is on.

**Syntax** RevealCodes (State: *Numeric Equivalent* or *Enumerated Type*)

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to turn on Reveal Codes. If you omit the parameter, this command acts as a toggle.

0 Off!  
1 On!

**Route** View, Reveal Codes.

**Example** To turn off Reveal Codes, the command is:

RevealCodes (State: Off!)

---

# RulerBarShow

**Purpose** RulerBarShow specifies whether to display the Ruler Bar.

**Syntax** RulerBarShow (State: *Numeric Equivalent* or *Enumerated Type*)

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display the Ruler Bar. If you omit the parameter, this command acts as a toggle.

0 Off!  
1 On!

**Route** View, Ruler Bar.

**Example** To display the Ruler Bar, the command is:

RulerBarShow (State: Yes!)

---

## SaveAll

|                |                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SaveAll saves all unsaved documents.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | SaveAll ()                                                                                                                               |

---

## ScrollScreenLeft

|                 |                                                                       |
|-----------------|-----------------------------------------------------------------------|
| <b>Purpose</b>  | ScrollScreenLeft displays the hidden left side of a document window.  |
| <b>Syntax</b>   | ScrollScreenLeft ()                                                   |
| <b>Route</b>    | Press Ctrl+PgUp or click the left arrow of the horizontal scroll bar. |
| <b>See Also</b> | ♦ ScrollScreenRight                                                   |

---

## ScrollScreenRight

|                 |                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ScrollScreenRight displays the hidden right side of a document window.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | ScrollScreenRight ()                                                                                                                                                       |
| <b>See Also</b> | ♦ ScrollScreenLeft                                                                                                                                                         |

---

## SearchCaseSensitive

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| <b>Purpose</b> | SearchCaseSensitive specifies whether Find is case sensitive.                     |
| <b>Syntax</b>  | SearchCaseSensitive (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether a search is case sensitive.

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

---

**Route** Edit, Find, enter search text, Match, Case, Find Next or Find Prev.

---

**Example** To specify a case-sensitive search, the command is:

```
SearchCaseSensitive (State: Yes!)
```

---

**See Also** ♦ SearchString

---

## SearchFindWholeWordsOnly

---

**Purpose** SearchFindWholeWordsOnly specifies whether Find matches a whole word, or if a match includes only part of a word.

---

**Syntax** SearchFindWholeWordsOnly (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

---

**Route** Edit, Find or Replace, Match, Whole Word.

---

**Example** To find whole words only, the command is:

```
SearchFindWholeWordsOnly (State: Yes!)
```

---

## SearchInSelection

---

**Purpose** SearchInSelection specifies whether search is restricted to selected text.

---

**Syntax** SearchInSelection (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type

- 0 No!
- 1 Yes!

---

**Route** Edit, Find or Replace, Options, Limit Find Within Selection.

---

**Example** To restrict the search to a block of text, the command is:

SearchInSelection (State: Yes!)

---

## SearchNext

---

**Purpose** SearchNext executes a forward search.

---

**Syntax** SearchNext (SearchMode: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*SearchMode* Numeric Equivalent or Enumerated Type: Specifies the scope of a search.

- 0 Regular!
- 1 Extended!

---

**Route** Edit, Find, enter search text, Find Next.

---

**Example** To execute a forward extended search, the command is:

SearchNext (SearchMode: Extended!)

---

**See Also** ♦ SearchPrevious

---

## SearchPrevious

---

**Purpose** SearchPrevious executes a backward search.

---

**Syntax** SearchPrevious (SearchMode: *Numeric Equivalent* or *Enumerated Type*)



---

**Parameters**

*SearchMode* Numeric Equivalent or Enumerated Type: Specifies the scope of a search.

- 0 Regular!
- 1 Extended!

---

**Route** Edit, Find, enter search text, Find Prev.

---

**Example** To execute a backward extended search, the command is:

SearchPrevious (SearchMode: Extended!)

---

**See Also** ♦ SearchNext

---

## SearchString

---

**Purpose** SearchString specifies a string to search for.

IMPORTANT: This command must be followed by either SearchNext or SearchPrevious.

---

**Syntax** SearchString (StrgToLookFor: *Character Expression*)

---

**Parameters**

*StrgToLookFor* Character Expression: Specifies search text and/or codes.

---

**Route** Edit, Find or Replace, enter search text.

---

**Example** To search for the next occurrence of NOTE, the command is:

SearchString (StrgToLookFor: "NOTE")  
SearchNext (SearchMode: Extended!)

---

## SearchWrap

---

**Purpose** SearchWrap specifies whether Find searches from the beginning of a document after it reaches the end of a document.

---

**Syntax** SearchWrap (State: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type  
0 No!  
1 Yes!

---

**Route** Edit, Find or Replace, Options, Wrap at Beg./End of Document.

---

**Example** To turn on SearchWrap, the command is:  
SearchWrap (State: Yes!)

---

---

## SecondaryPageNumber

**Purpose** SecondaryPageNumber specifies a secondary page number and inserts the code into a document.

---

**Syntax** SecondaryPageNumber (Page: *Numeric Expression*)

---

**Parameters**

*Page* Numeric Expression: Specifies a new secondary page number.

---

**Route** Layout, Page, Numbering, Value, New Secondary Number.

---

**Example** To define a secondary page number as 10, the command is:  
SecondaryPageNumber (Page: 10)

---

**See Also** ♦ PageNumber ♦ ?SecondaryPageNumber

---

---

## SecondaryPageNumberDecrement

**Purpose** SecondaryPageNumberDecrement decreases the value of the current secondary page number.

---

**Syntax** SecondaryPageNumberDecrement ()

---

**Route** Layout, Page, Numbering, Value, Increase/Decrease Existing Secondary Number.

---

---

## SecondaryPageNumberDisplay

---

|                |                                                                                                                                            |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SecondaryPageNumberDisplay displays the current secondary page number, which is entered as a code and updated as the document is modified. |
| <b>Syntax</b>  | SecondaryPageNumberDisplay ()                                                                                                              |
| <b>Route</b>   | Layout, Page, Numbering, Value, Insert and Display at Insertion Point.                                                                     |

---

## SecondaryPageNumberIncrement

---

|                |                                                                                        |
|----------------|----------------------------------------------------------------------------------------|
| <b>Purpose</b> | SecondaryPageNumberIncrement increases the value of the current secondary page number. |
| <b>Syntax</b>  | SecondaryPageNumberIncrement ()                                                        |
| <b>Route</b>   | Layout, Page, Numbering, Value, Increase/Decrease Existing Secondary Number.           |

---

## SecondaryPageNumberMethod

---

|                        |                                                                                                                                                                                                                                                           |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>         | SecondaryPageNumberMethod specifies a numbering method for secondary page numbers.                                                                                                                                                                        |
| <b>Syntax</b>          | SecondaryPageNumberMethod (NumberingMethod: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                         |
| <b>Parameters</b>      |                                                                                                                                                                                                                                                           |
| <i>NumberingMethod</i> | Numeric Equivalent or Enumerated Type: Specifies a numbering method for secondary page numbers.<br><ul style="list-style-type: none"><li>0 Numbers!</li><li>1 LowerLetters!</li><li>2 UpperLetters!</li><li>3 LowerRoman!</li><li>4 UpperRoman!</li></ul> |
| <b>Route</b>           | Layout, Page, Numbering, Options button, Secondary, select option.                                                                                                                                                                                        |

---

---

**Example** To display secondary page numbers in uppercase roman numerals, the command is:  
SecondaryPageNumberMethod (NumberingMethod: UpperRoman!)

---

**See Also** ♦ ?SecondaryPageNumberMethod

---

---

## SelectAll

---

**Purpose** SelectAll selects all text and graphics in the current window.

---

**Syntax** SelectAll ()

---

**Route** Edit, Select, All.

---

---

## SelectCell

---

**Purpose** SelectCell selects the cell with the insertion point, including text and attributes.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** SelectCell ()

---

**See Also** ♦ TableSelectOn

---

---

## SelectCellDown

---

**Purpose** SelectCellDown moves the insertion point down one row, selecting text and codes from the previous position to the new position.  
IMPORTANT: This command does not move the insertion point below a table when the insertion point is in the bottom row. If the insertion point is not in a table, this command ends the macro or redirects macro execution to the LABEL command specified by ONERROR.

---

**Syntax** SelectCellDown ()

---

---

|                 |                                      |
|-----------------|--------------------------------------|
| <b>Route</b>    | Press Alt+Shift+Down Arrow.          |
| <b>See Also</b> | ♦ SelectCellUp ♦ SelectCellDownArrow |

---

## SelectCellDownArrow

---

|                 |                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectCellDownArrow moves the insertion point, selecting text and codes between the old and new insertion points.<br><br>IMPORTANT: This command moves the insertion point below a table when the insertion point is in the bottom row.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectCellDownArrow ()                                                                                                                                                                                                                                                                                                                          |
| <b>See Also</b> | ♦ SelectCellUp ♦ SelectCellDown                                                                                                                                                                                                                                                                                                                 |

---

## SelectCellLeft

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectCellLeft moves the insertion point left one column, selecting text and codes from the previous position to the new position.<br><br>IMPORTANT: If the insertion point is already in the farthest left column, this command moves the insertion point up one row to the farthest left column. If the insertion point is outside a table, this command ends the macro or redirects macro execution to the LABEL specified by ONERROR.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectCellLeft ()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>See Also</b> | ♦ SelectCellRight                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

---

---

## SelectCellRight

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectCellRight moves the insertion point right one column, selecting text and codes from the previous position to the new position.<br><br>IMPORTANT: If the insertion point is already in the farthest right column, this command moves the insertion point down one row to the farthest left column. If the insertion point is outside a table, this command ends the macro or redirects macro execution to the LABEL specified by ONERROR.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectCellRight ()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>See Also</b> | ♦ SelectCellLeft                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

---

## SelectCellUp

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectCellUp moves the insertion point up one row, selecting text and codes from the previous position to the new position.<br><br>IMPORTANT: This command does not move the insertion point above a table when the insertion point is in the top row of a table. If the insertion point is not in a table, this command ends the macro or redirects macro execution to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | SelectCellUp ()                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Route</b>    | Press Alt+Shift+Up Arrow.                                                                                                                                                                                                                                                                                                                                                                               |
| <b>See Also</b> | ♦ SelectCellDown ♦ SelectCellUpArrow                                                                                                                                                                                                                                                                                                                                                                    |

---

## SelectCellUpArrow

---

|                |                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectCellUpArrow moves the insertion point up one row, selecting text and codes from the previous position to the new position.<br><br>IMPORTANT: This command moves the insertion point above a table when the insertion point is in the top row of a table.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                      |
|-----------------|--------------------------------------|
| <b>Syntax</b>   | SelectCellUpArrow ()                 |
| <b>See Also</b> | ♦ SelectCellDownArrow ♦ SelectCellUp |

---

## SelectCharNext

---

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectCharNext selects text or code one character position to the right. |
| <b>Syntax</b>   | SelectCharNext ()                                                        |
| <b>Route</b>    | Press Shift+Right Arrow.                                                 |
| <b>See Also</b> | ♦ SelectCharPrevious                                                     |

---

## SelectCharPrevious

---

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectCharPrevious selects text and codes one character position to the left. |
| <b>Syntax</b>   | SelectCharPrevious ()                                                         |
| <b>Route</b>    | Press Shift+Left Arrow.                                                       |
| <b>See Also</b> | ♦ SelectCharNext                                                              |

---

## SelectColumnBottom

---

|                 |                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectColumnBottom moves the insertion point to the bottom of the current column, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectColumnBottom ()                                                                                                                                      |
| <b>Route</b>    | Press Shift+Alt+End.                                                                                                                                       |
| <b>See Also</b> | ♦ SelectColumnTop                                                                                                                                          |

---

---

## SelectColumnNext

---

**Purpose**

SelectColumnNext moves the insertion point right one column, selecting text and codes from the previous position to the new position.

IMPORTANT: If the insertion point is in the farthest right column, this command moves the insertion point to the beginning of the current line, and selects text and codes from the previous position to the new position.

**Syntax**

SelectColumnNext ()

**Route**

Press Shift+Alt+Right Arrow.

**See Also**

♦ SelectColumnPrevious

---

## SelectColumnPrevious

---

**Purpose**

SelectColumnPrevious moves the insertion point left one column, selecting text and codes from the previous position to the new position.

IMPORTANT: If the insertion point is in the farthest left column, this command moves the insertion point to the beginning of the current line, and selects text and codes from the previous position to the new position.

**Syntax**

SelectColumnPrevious ()

**Route**

Press Shift+Alt+Left Arrow.

**See Also**

♦ SelectColumnNext

---

## SelectColumnTop

---

**Purpose**

SelectColumnTop moves the insertion point to the top of the current column, selecting text and codes from the previous position to the new position.

**Syntax**

SelectColumnTop ()

**Route**

Press Shift+Alt+Home.

---



---

**See Also** ♦ SelectColumnBottom

---

## SelectDelete

---

**Purpose** SelectDelete deletes selected text.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** SelectDelete ()

---

## SelectDocBottom

---

**Purpose** SelectDocBottom moves the insertion point to the end of the document, selecting text and codes from the previous position to the new position.

---

**Syntax** SelectDocBottom ()

---

**Route** Press Shift+Ctrl+End.

---

**See Also** ♦ SelectDocTop

---

## SelectDocTop

---

**Purpose** SelectDocTop moves the insertion point to the beginning of the document, selecting text and codes from the previous position to the new position.

---

**Syntax** SelectDocTop ()

---

**Route** Press Shift+Ctrl+Home.

---

**See Also** ♦ SelectDocBottom ♦ SelectDocVeryTop

---

---

## SelectDocVeryTop

|                 |                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectDocVeryTop moves the insertion point to the very beginning of the document, before all codes, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectDocVeryTop ()                                                                                                                                                          |
| <b>Route</b>    | Press Ctrl+Shift+Home twice.                                                                                                                                                 |
| <b>See Also</b> | ♦ SelectDocBottom ♦ SelectDocTop                                                                                                                                             |

---

## SelectLineBegin

|                 |                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectLineBegin moves the insertion point to the beginning of the current line, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectLineBegin ()                                                                                                                                       |
| <b>Route</b>    | Press Shift+Home.                                                                                                                                        |
| <b>See Also</b> | ♦ SelectLineEnd                                                                                                                                          |

---

## SelectLineDown

|                 |                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectLineDown moves the insertion point down one line, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectLineDown ()                                                                                                                |
| <b>Route</b>    | Press Shift+Down Arrow.                                                                                                          |
| <b>See Also</b> | ♦ SelectLineUp                                                                                                                   |

---

## SelectLineEnd

|                 |                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectLineEnd moves the insertion point to the end of the current line, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectLineEnd ()                                                                                                                                 |
| <b>Route</b>    | Press Shift+End.                                                                                                                                 |
| <b>See Also</b> | ♦ SelectLineBegin                                                                                                                                |

---

## SelectLineUp

|                 |                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectLineUp moves the insertion point up one line, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectLineUp ()                                                                                                              |
| <b>Route</b>    | Press Shift+Up Arrow.                                                                                                        |
| <b>See Also</b> | ♦ SelectLineDown                                                                                                             |

---

## SelectLineVeryBegin

|                 |                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectLineVeryBegin moves the insertion point to the beginning of the current line, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectLineVeryBegin ()                                                                                                                                       |
| <b>Route</b>    | Press Shift+Home Home.                                                                                                                                       |
| <b>See Also</b> | ♦ PosLineVeryBeg ♦ SelectLineBegin ♦ SelectLineVeryEnd                                                                                                       |

---

## SelectLineVeryEnd

---

|                 |                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectLineVeryEnd moves the insertion point to the end of the current line, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectLineVeryEnd ()                                                                                                                                 |
| <b>Route</b>    | Press Shift+End.                                                                                                                                     |
| <b>See Also</b> | ♦ PosLineVeryEnd ♦ SelectLineEnd ♦ SelectLineBegin ♦ SelectLineVeryBegin                                                                             |

---

## SelectMode

---

|                   |                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | SelectMode specifies whether to turn on Select. If no parameter is specified, this command toggles Select. The regular positioning keys extend a selection. |
| <b>Syntax</b>     | SelectMode (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                    |
| <b>Parameters</b> |                                                                                                                                                             |
| <i>State</i>      | Numeric Equivalent or Enumerated Type<br>0 Off!<br>1 On!                                                                                                    |
| <b>Route</b>      | Press F8.                                                                                                                                                   |
| <b>Example</b>    | To turn on Select, the command is:<br>SelectMode (State: On!)                                                                                               |

---

## SelectOff

---

|                |                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectOff turns off Select.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | SelectOff ()                                                                                                                        |

---

---

## SelectOn

---

**Purpose** SelectOn turns on Select.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** SelectOn (SelectMode: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*SelectMode* Numeric Equivalent or Enumerated Type: Specifies a selection type.

- 1 CharMode!
- 2 WordMode!
- 4 SentenceMode!
- 8 ParagraphMode!
- 16 PageMode!
- 32 RectangleMode!
- 64 TabularColumnMode!
- 128 DocMode!
- 256 CharNoCellMode!

---

**Example** To select the current sentence, the command is:

SelectOn (SelectMode: SentenceMode!)

---

## SelectPage

---

**Purpose** SelectPage selects the current page.

---

**Syntax** SelectPage ()

---

**Route** Edit, Select, Page.

---

## SelectPageNext

---

**Purpose** SelectPageNext moves the insertion point to the beginning of the next page, selecting text and codes from the previous position to the new position. If the insertion point is already on the last page, this command moves the insertion point to the end of the document, selecting text and codes from the previous position to the new position.

---

|                 |                       |
|-----------------|-----------------------|
| <b>Syntax</b>   | SelectPageNext ()     |
| <b>Route</b>    | Press Shift+Alt+PgDn. |
| <b>See Also</b> | ♦ SelectPagePrevious  |

---

## SelectPagePrevious

---

|                 |                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectPagePrevious moves the insertion point to the beginning of the previous page, selecting text and codes from the previous position to the new position. If the insertion point is already on the first page, this command moves the insertion point to the beginning of the document, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectPagePrevious ()                                                                                                                                                                                                                                                                                                                                               |
| <b>Route</b>    | Press Shift+Alt+PgUp.                                                                                                                                                                                                                                                                                                                                               |
| <b>See Also</b> | ♦ SelectPageNext                                                                                                                                                                                                                                                                                                                                                    |

---

## SelectParagraph

---

|                |                                                |
|----------------|------------------------------------------------|
| <b>Purpose</b> | SelectParagraph selects the current paragraph. |
| <b>Syntax</b>  | SelectParagraph ()                             |
| <b>Route</b>   | Edit, Select, Paragraph.                       |

---

## SelectParagraphNext

---

|                |                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectParagraphNext moves the insertion point to the beginning of the next paragraph, selecting text and codes from the previous position to the new position. If the insertion point is in the last paragraph of a document, this command moves the insertion point to the end of the document, selecting text and codes from the previous position to the new position. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

|                 |                              |
|-----------------|------------------------------|
| <b>Syntax</b>   | SelectParagraphNext ()       |
| <b>Route</b>    | Press Shift+Ctrl+Down Arrow. |
| <b>See Also</b> | ♦ SelectParagraphPrevious    |

---

## SelectParagraphPrevious

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectParagraphPrevious moves the insertion point to the beginning of the current paragraph, selecting text and codes from the previous position to the new position. If the insertion point is at the beginning of a paragraph, this command moves the insertion point to the beginning of the preceding paragraph, selecting text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectParagraphPrevious ()                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Route</b>    | Press Shift+Ctrl+Up Arrow.                                                                                                                                                                                                                                                                                                                                                                    |
| <b>See Also</b> | ♦ SelectParagraphNext                                                                                                                                                                                                                                                                                                                                                                         |

---

## SelectRectangle

---

|                |                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectRectangle selects a blocked rectangle.<br><br>IMPORTANT: SelectMode must be on (press F8) for this option to record. You can then use regular positioning keys to extend a selection. |
| <b>Syntax</b>  | SelectRectangle ()                                                                                                                                                                          |
| <b>Route</b>   | Edit, Select, Rectangle.                                                                                                                                                                    |

---

## SelectSave

---

|                |                                                     |
|----------------|-----------------------------------------------------|
| <b>Purpose</b> | SelectSave saves selected text as a new file.       |
| <b>Syntax</b>  | SelectSave (Filename: <i>Character Expression</i> ) |

---

---

**Parameters**

*Filename* Character Expression: Specifies a name for the new file containing the selection.

---

**Route** Select text, File, Save or Save As, Selected Text, OK, enter a filename.

---

**Example** To save selected text as a file named DOC1.WPD, the command is:

SelectSave (Filename: "DOC1.WPD")

---

## SelectScreenDown

---

**Purpose** SelectScreenDown selects text and codes from the insertion point to the bottom of the screen.

IMPORTANT: If the insertion point is already at the bottom of the screen, this command positions the insertion point at the bottom of the next screen, and selects text and codes from the previous position to the new position.

---

**Syntax** SelectScreenDown ()

---

**Route** Press Shift+PgDn.

---

**See Also** ♦ SelectScreenUp

---

## SelectScreenLeft

---

**Purpose** SelectScreenLeft positions the insertion point on the left side of the screen, and selects text and codes from the previous position to the new position.

IMPORTANT: If the insertion point is already on the left side of the screen, this command positions the insertion point on the left side of the next screen, or at the left margin of a document.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** SelectScreenLeft ()

---

**See Also** ♦ PosScreenLeft ♦ SelectScreenRight



---

## SelectScreenRight

---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectScreenRight positions the insertion point on the right side of the screen, and selects text and codes from the previous position to the new position.<br><br>IMPORTANT: If the insertion point is already on the right side of the screen, this command positions the insertion point on the right side of the next screen, or at the right margin of a document.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectScreenRight ()                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>See Also</b> | ♦ PosScreenRight ♦ SelectScreenLeft                                                                                                                                                                                                                                                                                                                                                                                                                                             |

---

## SelectScreenUp

---

|                 |                                                                                                                                                                                                                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectScreenUp selects text and codes from the insertion point to the top of the screen.<br><br>IMPORTANT: If the insertion point is already at the top of the screen, this command positions the insertion point at the top of the next screen, and selects text and codes from the previous position to the new position. |
| <b>Syntax</b>   | SelectScreenUp ()                                                                                                                                                                                                                                                                                                           |
| <b>Route</b>    | Press Shift+PgUp.                                                                                                                                                                                                                                                                                                           |
| <b>See Also</b> | ♦ SelectScreenDown                                                                                                                                                                                                                                                                                                          |

---

## SelectSentence

---

|                |                                              |
|----------------|----------------------------------------------|
| <b>Purpose</b> | SelectSentence selects the current sentence. |
| <b>Syntax</b>  | SelectSentence ()                            |
| <b>Route</b>   | Edit, Select, Sentence.                      |

---

---

## SelectSentenceNext

---

|                 |                                                                                                                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectSentenceNext selects everything from the insertion point to the beginning of the next sentence.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectSentenceNext ()                                                                                                                                                                                         |
| <b>See Also</b> | ♦ SelectSentencePrevious                                                                                                                                                                                      |

---

## SelectSentencePrevious

---

|                 |                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectSentencePrevious selects everything from the insertion point to the end of the previous sentence.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectSentencePrevious ()                                                                                                                                                                                       |
| <b>See Also</b> | ♦ SelectSentenceNext                                                                                                                                                                                            |

---

## SelectTable

---

|                 |                                                                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectTable selects all cells in the current table.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.<br><br>IMPORTANT: If the insertion point is not in a table, this command terminates the macro or redirects the macro to the LABEL specified by ONERROR. |
| <b>Syntax</b>   | SelectTable ()                                                                                                                                                                                                                                                                                                      |
| <b>See Also</b> | ♦ SelectTableColumn ♦ SelectTableRow                                                                                                                                                                                                                                                                                |

---

---

## SelectTableColumn

|                 |                                                                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectTableColumn selects all cells in the current column.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectTableColumn ()                                                                                                                                               |
| <b>See Also</b> | ♦ SelectTable ♦ SelectTableRow                                                                                                                                     |

---

## SelectTableColumnExtendLeft

|                 |                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectTableColumnExtendLeft selects cells from the insertion point to the farthest left cell.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectTableColumnExtendLeft ()                                                                                                                                                                        |
| <b>See Also</b> | ♦ SelectTableColumnExtendRight                                                                                                                                                                        |

---

## SelectTableColumnExtendRight

|                 |                                                                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectTableColumnExtendRight selects all cells from the insertion point to the beginning of the farthest right cell.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectTableColumnExtendRight ()                                                                                                                                                                                              |
| <b>See Also</b> | ♦ SelectTableColumnExtendLeft                                                                                                                                                                                                |

---

## SelectTableRow

|                 |                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SelectTableRow selects the cells in the current row.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SelectTableRow ()                                                                                                                                        |
| <b>See Also</b> | ♦ SelectTable ♦ SelectTableColumn                                                                                                                        |

---

## SelectTabularColumn

|                |                                                                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectTabularColumn selects one or more blocked tabular columns.<br>IMPORTANT: SelectMode must be on (press F8) for this option to be recordable. You can then use regular positioning keys to extend a selection. |
| <b>Syntax</b>  | SelectTabularColumn ()                                                                                                                                                                                             |
| <b>Route</b>   | Edit, Select, Tabular Column.                                                                                                                                                                                      |

---

## SelectToggle

|                |                                                                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectToggle toggles Select mode.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | SelectToggle ()                                                                                                                       |

---

## SelectWord

|                |                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectWord selects the word containing the insertion point.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |               |
|---------------|---------------|
| <b>Syntax</b> | SelectWord () |
|---------------|---------------|

---

## SelectWordNext

|                |                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectWordNext selects everything from the insertion point to the beginning of the next word. |
|----------------|-----------------------------------------------------------------------------------------------|

|               |                   |
|---------------|-------------------|
| <b>Syntax</b> | SelectWordNext () |
|---------------|-------------------|

|              |                               |
|--------------|-------------------------------|
| <b>Route</b> | Press Shift+Ctrl+Right Arrow. |
|--------------|-------------------------------|

---

## SelectWordPrevious

|                |                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SelectWordPrevious selects everything from the insertion point to the end of the previous word. |
|----------------|-------------------------------------------------------------------------------------------------|

|               |                       |
|---------------|-----------------------|
| <b>Syntax</b> | SelectWordPrevious () |
|---------------|-----------------------|

|              |                              |
|--------------|------------------------------|
| <b>Route</b> | Press Shift+Ctrl+Left Arrow. |
|--------------|------------------------------|

---

## SetDefaultFileFormat

|                |                                                         |
|----------------|---------------------------------------------------------|
| <b>Purpose</b> | SetDefaultFileFormat specifies the default file format. |
|----------------|---------------------------------------------------------|

|               |                                                                                          |
|---------------|------------------------------------------------------------------------------------------|
| <b>Syntax</b> | SetDefaultFileFormat (DefaultType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|------------------------------------------------------------------------------------------|

### Parameters

|                    |                                       |
|--------------------|---------------------------------------|
| <i>DefaultType</i> | Numeric Equivalent or Enumerated Type |
|--------------------|---------------------------------------|

- 1 WordPerfect\_42!
- 2 WordPerfect\_50!
- 3 WordPerfect\_51!
- 4 WordPerfect\_60!
- 30 WordStar\_33!
- 31 WordStar\_331!
- 32 WordStar\_34!
- 33 WordStar\_40!
- 34 WordStar\_50!
- 35 WordStar\_55!

36 WordStar\_60!  
50 WordStar2000\_10!  
51 WordStar2000\_20!  
52 WordStar2000\_30!  
62 MicroSoftWord\_40!  
63 MicroSoftWord\_50!  
64 MicroSoftWord\_55!  
70 WordForWindows\_10!  
71 WordForWindows\_11!  
72 WordForWindows\_1A!  
73 WordForWindows\_20!  
74 WordForWindows\_20a!  
75 WordForWindows\_20b!  
80 RFT!  
81 FFT!  
82 DisplayWrite\_40!  
83 DisplayWrite\_42!  
84 DisplayWrite\_50!  
100 MultiMate\_33!  
101 MultiMateAdvantage\_36!  
102 MultiMateAdvantageII\_10!  
103 MultiMate\_40!  
120 OfficeWriter\_60!  
121 OfficeWriter\_61!  
122 OfficeWriter\_611!  
123 OfficeWriter\_62!  
150 RTF!  
160 ASCIIStandard!  
161 ANSIText\_Windows!  
163 DelimitedText!  
165 ASCIIStripped!  
173 Kermit\_7bit\_Transfer!  
175 NavyDIF!  
190 XyWriteIIIPlus\_355!  
191 XyWriteIIIPlus\_356!  
311 AmiPro\_12!  
312 AmiPro\_12A!  
313 AmiPro\_12B!  
314 AmiPro\_20!  
315 AmiPro\_30!  
656 ANSIDelimitedTextWindows!  
657 ANSISGenericWordProcessor\_Windows!  
807 Excel\_40!

---

**Route**

File, Save As, Setup button, select a file format from the Default Save File Format list box.

|                 |                                                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b>  | To set the default file format to WordPerfect 6.0 for Windows, the command is:<br><code>SetDefaultFileFormat (DefaultType: WordPerfect_60!)</code> |
| <b>See Also</b> | ♦ PrefSave                                                                                                                                         |

---

## SetDocumentMapping

|                   |                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | SetDocumentMapping specifies which character map contains one-byte characters, as opposed to extended characters, when additional character maps are installed.                     |
| <b>Syntax</b>     | SetDocumentMapping (MapName: <i>Character Expression</i> ; Location: <i>Numeric Equivalent or Enumerated Type</i> ; Filename: <i>Character Expression</i> )                         |
| <b>Parameters</b> |                                                                                                                                                                                     |
| <i>MapName</i>    | Character Expression: Specifies a character map name.                                                                                                                               |
| <i>Location</i>   | Numeric Equivalent or Enumerated Type: Specifies whether a character map is internal or external.<br>0 MapInternal!<br>1 MapExternal!                                               |
| <i>Filename</i>   | Character Expression: Specifies a file containing a character map.                                                                                                                  |
| <b>Route</b>      | Layout, Document, Character Mapping, select a character map from the Document Character Map list box, Apply.                                                                        |
| <b>Example</b>    | To specify an internal Greek character map as the map containing one-byte characters, the command is:<br><code>SetDocumentMapping (MapName: "Greek"; Location: MapInternal!)</code> |

---

## SettingsDlg

|                |                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SettingsDlg displays the Preferences dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | SettingsDlg ()                                                                                                                                           |

---

## SetupWritingToolsDlg

---

|                |                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SetupWritingToolsDlg displays the Writing Tools dialog box.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | SetupWritingToolsDlg ()                                                                                                                                         |

---

## ShiftTabKey

---

|                |                                                           |
|----------------|-----------------------------------------------------------|
| <b>Purpose</b> | ShiftTabKey inserts [Hd Back Tab] at the insertion point. |
| <b>Syntax</b>  | ShiftTabKey ()                                            |
| <b>Route</b>   | Press Shift+Tab.                                          |

---

## ShowBoxes

---

|                   |                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ShowBoxes specifies whether to display graphics boxes.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | ShowBoxes (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                    |
| <b>Parameters</b> |                                                                                                                                                            |
| <i>State</i>      | Numeric Equivalent or Enumerated Type<br>0 Off!<br>1 On!                                                                                                   |
| <b>Example</b>    | To display graphics boxes, the command is:<br>ShowBoxes (State: On!)                                                                                       |
| <b>See Also</b>   | ♦ ?ShowBoxes                                                                                                                                               |

---



---

## ShowGraphics

---

**Purpose** ShowGraphics toggles Graphics.

---

**Syntax** ShowGraphics ()

---

**Route** View, Graphics.

---

## ShowHiddenText

---

**Purpose** ShowHiddenText toggles Hidden Text.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** ShowHiddenText ()

---

---

## ShowSymbols

---

**Purpose** ShowSymbols toggles ShowSymbols for features such as Space, Hard Return, and Tab.

---

**Syntax** ShowSymbols ()

---

**Route** View, Show ¶.

---

---

## SoftHyphen

---

**Purpose** SoftHyphen inserts a soft hyphen at the insertion point.

---

**Syntax** SoftHyphen ()

---

**Route** Layout, Line, Other Codes, Soft Hyphen [-Soft Hyphen].

---

---

## SoftSpace

|                 |                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SoftSpace inserts a space at the insertion point.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | SoftSpace ()                                                                                                                                          |
| <b>See Also</b> | ♦ HardSpace                                                                                                                                           |

---

## Sort

|                   |                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | Sort specifies a file to sort and a file to which the changes are written. The current document is the default.               |
| <b>Syntax</b>     | Sort (InFile: <i>Character Expression</i> ; OutFile: <i>Character Expression</i> )                                            |
| <b>Parameters</b> |                                                                                                                               |
| <i>InFile</i>     | Character Expression: Specifies a file to sort. If this parameter is omitted, the current document is sorted.                 |
| <i>OutFile</i>    | Character Expression: Specifies a file a sort writes to. If this parameter is omitted, the sort updates the current document. |
| <b>Route</b>      | Tools, Sort.                                                                                                                  |
| <b>Example</b>    | To sort the current document and save it as a file named RESULT, the command is:<br>Sort (OutFile: "RESULT")                  |

---

## SortAction

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | SortAction specifies whether to sort a file or select only certain records.               |
| <b>Syntax</b>     | SortAction (SortAction: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )             |
| <b>Parameters</b> |                                                                                           |
| <i>SortAction</i> | Numeric Equivalent or Enumerated Type: Specifies a sort type.<br>0 Sort!<br>1 SelectOnly! |

---

|                 |                                                                                                                                   |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Tools, Sort, Select Without Sorting.                                                                                              |
| <b>Example</b>  | To select only certain records without changing their order, the command is:<br><code>SortAction (SortAction: SelectOnly!)</code> |
| <b>See Also</b> | ♦ ?SortGetAction                                                                                                                  |

---

## SortCaseOrder

---

|                   |                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | SortCaseOrder specifies case priority.                                                                                   |
| <b>Syntax</b>     | SortCaseOrder (Case: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                               |
| <b>Parameters</b> |                                                                                                                          |
| <i>Case</i>       | Numeric Equivalent or Enumerated Type: Specifies which case has sort priority.<br>0 LowercaseFirst!<br>1 UppercaseFirst! |
| <b>Route</b>      | Tools, Sort, Uppercase First.                                                                                            |
| <b>Example</b>    | To prioritize uppercase entries, the command is:<br><code>SortCaseOrder (Case: UppercaseFirst!)</code>                   |
| <b>See Also</b>   | ♦ ?SortGetCase                                                                                                           |

---

## SortDlg

---

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | SortDlg displays the Sort (Source and Destination) dialog box. |
| <b>Syntax</b>  | SortDlg ()                                                     |
| <b>Route</b>   | Tools, Sort.                                                   |

---

---

# SortKeys

|                   |                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | SortKeys specifies sort key options.                                                                                                                                                                                                                |
| <b>Syntax</b>     | SortKeys ({Field: <i>Numeric Expression</i> ; Line: <i>Numeric Expression</i> ; Word: <i>Numeric Expression</i> ; SortType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; SortOrder: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> }) |
| <b>Parameters</b> |                                                                                                                                                                                                                                                     |
| <i>Field</i>      | Numeric Expression: Specifies a field to sort.                                                                                                                                                                                                      |
| <i>Line</i>       | Numeric Expression: Specifies a line to sort within a specified field. This parameter is always one (1) if the sort type is Line.                                                                                                                   |
| <i>Word</i>       | Numeric Expression: Specifies a word to sort within a specified line and field.                                                                                                                                                                     |
| <i>SortType</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to perform a numeric or alphanumeric sort.<br>0 Alphanumeric!<br>1 Numeric!                                                                                                                |
| <i>SortOrder</i>  | Numeric Equivalent or Enumerated Type: Specifies whether to sort a-z and 0-9 or z-a and 9-0.<br>0 Ascending!<br>1 Descending!                                                                                                                       |
| <b>Route</b>      | Tools, Sort, specify key definitions.                                                                                                                                                                                                               |
| <b>Example</b>    | To sort ZIP Codes that are the only word on the only line in the sixth field of a data file, the command is:<br><br>SortKeys ({ Field: 6; Line: 1; Word: 1; SortType: Numeric!; SortOrder: Ascending! })                                            |

---

# SortSelectString

|                |                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SortSelectString inserts a character string into the Select Records text box of the Sort dialog box. |
| <b>Syntax</b>  | SortSelectString (SelectString: <i>Character Expression</i> )                                        |

---

**Parameters**

*SelectString* Character Expression: Specifies text to select.

---

**Route** Tools, Sort, Select Records, type character string.

---

**Example** To sort and select only records that contain the ZIP Code 84057 in the field defined by Key1, the command is:

SortSelectString (SelectString: "Key1=84057")

---

## SortType

---

**Purpose** SortType specifies a source file format.

IMPORTANT: ParagraphSort! sorts the file leaving paragraphs intact; LineSort! sorts each line of the file individually; ColumnSort! sorts parallel columns; TableSort! sorts a table where rows act as lines and cells act as fields; and MergeSort! sorts a data file.

---

**Syntax** SortType (SortType: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*SortType* Numeric Equivalent or Enumerated Type: Specifies a sort type reflecting a source file format.

- 0 ParagraphSort!
- 1 LineSort!
- 2 ColumnSort!
- 3 TableSort!
- 4 MergeSort!

---

**Route** Tools, Sort, select a record type.

---

**Example** To specify the source file as a data file, the command is:

SortType (SortType: MergeSort!)

---

## SoundClipCreate

---

**Purpose** SoundClipCreate inserts a sound clip from a file into Sound Clips in the Document dialog box.

---

**Syntax** SoundClipCreate (Description: *Character Expression*; Filename: *Character Expression*; SoundFileType: *Numeric Equivalent* or *Enumerated Type*; ClipType: *Numeric Equivalent* or *Enumerated Type*; LeftVolume: *Numeric Expression*; RightVolume: *Numeric Expression*)

---

**Parameters**

*Description* Character Expression: Specifies a sound clip description.

*Filename* Character Expression: Specifies a sound clip filename.

*SoundFileType* Numeric Equivalent or Enumerated Type: Specifies a sound clip location.

0 OnDisk!  
1 InDoc!

*ClipType* Numeric Equivalent or Enumerated Type: Specifies a sound clip type.

0 MIDI!  
1 DigitalAudio!

*LeftVolume* Numeric Expression: Specifies the left speaker volume.

*RightVolume* Numeric Expression: Specifies the right speaker volume.

---

**Route** Insert, Sound, Insert, enter a filename.

---

**Example** To give a sound clip on disk the description Duplicate of Exit and insert it into the document, the command is:

SoundClipCreate (Description: "Duplicate of Exit"; Filename: "C:\WP60\EXIT.SND"; SoundFileType: OnDisk!; ClipType: DigitalAudio!; LeftVolume: 10; RightVolume: 10)

---

## SoundClipDelete

---

**Purpose** SoundClipDelete deletes a sound clip.

---

**Syntax** SoundClipDelete (Description: *Character Expression*)

---

**Parameters**

*Description* Character Expression: Specifies a sound clip description to delete.

---

**Route** Insert, Sound, select a sound clip, Delete.

---

**Example** To delete a sound clip named OPEN.SND, the command is:

SoundClipDelete (Description: "OPEN.SND")

---

---

## SoundClipDlg

---

**Purpose** SoundClipDlg displays the Sound Clips dialog box.

---

**Syntax** SoundClipDlg ()

---

**Route** Insert, Sound.

---

---

## SoundClipPlayNext

---

**Purpose** SoundClipPlayNext plays a specified sound clip.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** SoundClipPlayNext (Description: *Character Expression*; StartPos: *Numeric Expression*; EndPos: *Numeric Expression*)

---

### Parameters

*Description* Character Expression: Specifies the description of a sound clip to play.

*StartPos* Numeric Expression: Specifies in seconds a sound clip starting point.

*EndPos* Numeric Expression: Specifies in seconds a sound clip ending point.

---

**Example** To play a sound clip with the description Close, the command is:

SoundClipPlayNext (Description: "Close"; StartPos: 1; EndPos: 5)

---

## SoundClipSetAttributes

---

**Purpose** SoundClipSetAttributes specifies sound clip attributes.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** SoundClipSetAttributes (Description: *Character Expression*; Type: *Numeric Equivalent* or *Enumerated Type*; LeftVolume: *Numeric Expression*; RightVolume: *Numeric Expression*)

---

**Parameters**

*Description* Character Expression: Specifies a sound clip description.

*Type* Numeric Equivalent or Enumerated Type

- 0 MIDI!
- 1 DigitalAudio!

*LeftVolume* Numeric Expression: Specifies left speaker volume.

*RightVolume* Numeric Expression: Specifies right speaker volume.

---

**Example**

To specify sound clip attributes of a sound clip named Close, the command is:

SoundClipSetAttributes (Description: "Close"; Type: DigitalAudio!; LeftVolume: 10; RightVolume: 10)

---

## SoundClipSetDescription

---

**Purpose** SoundClipSetDescription renames a sound clip.

**Syntax** SoundClipSetDescription (OldDescription: *Character Expression*; NewDescription: *Character Expression*)

---

**Parameters**

*OldDescription* Character Expression: Specifies the old sound clip description name.

*NewDescription* Character Expression: Specifies the new sound clip description name.

---

**Route** Insert, Sound, select a sound clip, Edit Desc, enter a new description.

---

**Example**

To rename a sound clip, the command is:

SoundClipSetDescription (OldDescription: "CloseOld"; NewDescription: "CloseNew")

---

## SoundClipSetFilename

---

**Purpose** SoundClipSetFilename assigns a new filename to a sound clip.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.



---

**Syntax** SoundClipSetFilename (Description: *Character Expression*; Filename: *Character Expression*; Location: *Numeric Equivalent or Enumerated Type*)

---

**Parameters**

*Description* Character Expression: Specifies a sound clip description.

*Filename* Character Expression: Specifies a sound clip name.

*Location* Numeric Equivalent or Enumerated Type: Specifies a sound clip location.

0 OnDisk!  
1 InDoc!

---

**Example** To select a sound clip file from disk, the command is:

SoundClipSetFilename (Description: "Clip"; Filename: "C:\WPWIN\CLIP\WAV"; Location: OnDisk!)

---

---

## SpecificDeleteStringBegin

---

**Purpose** SpecificDeleteStringBegin deletes specific codes.

IMPORTANT: SpecificDeleteStringBegin must be followed by a macro command corresponding to a specific code to delete. This command must be used in conjunction with other commands, such as ReplaceBackward, ReplaceForward, and SpecificStringEnd.

---

**Syntax** SpecificDeleteStringBegin ()

---

**Route** Edit, Replace, Type, Specific Codes, select a code type, specify a specific code, Replace with Nothing, Replace or Replace All.

---

**See Also** ♦ ReplaceBackward ♦ ReplaceForward ♦ SpecificReplaceStringBegin ♦ SpecificSearchStringBegin ♦ SpecificStringEnd ♦ MatchSelection

---

---

## SpecificReplaceStringBegin

---

**Purpose** SpecificReplaceStringBegin replaces a specific code with another specific code.

IMPORTANT: A macro command corresponding to a specific replacement code must follow SpecificReplaceStringBegin. The code being replaced is specified using SpecificSearchStringBegin. SpecificReplaceStringBegin must be used in conjunction with other commands, such as ReplaceBackward, ReplaceForward, SpecificSearchStringBegin, and SpecificStringEnd.

---

|                 |                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------|
| <b>Syntax</b>   | SpecificReplaceStringBegin ()                                                                    |
| <b>Route</b>    | Edit, Replace, Type, Specific Codes, select a code type, specify a code, Replace or Replace All. |
| <b>See Also</b> | ♦ ReplaceBackward ♦ ReplaceForward ♦ SpecificSearchStringBegin<br>♦ SpecificStringEnd            |

---

## SpecificSearchStringBegin

|                 |                                                                                                                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | SpecificSearchStringBegin finds a specific code.<br><br>IMPORTANT: A macro command corresponding to a specific code to find must follow SpecificSearchStringBegin. This command must be used in conjunction with SpecificStringEnd. |
| <b>Syntax</b>   | SpecificSearchStringBegin ()                                                                                                                                                                                                        |
| <b>Route</b>    | Edit, Find, Type, Specific Codes, select a code type, specify a code, Find Next.<br>Edit, Replace, Type Specific Codes, select a code type, specify a code, Replace or Replace All.                                                 |
| <b>See Also</b> | ♦ SearchNext ♦ SearchPrevious ♦ SpecificDeleteStringBegin<br>♦ SpecificReplaceStringBegin ♦ SpecificStringEnd                                                                                                                       |

---

## SpecificStringEnd

|                |                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | SpecificStringEnd is the closing command for a specific code search. This command must be preceded by SpecificSearchStringBegin, SpecificReplaceStringBegin, or SpecificDeleteStringBegin, and a command which specifies the specific code to search. |
| <b>Syntax</b>  | SpecificStringEnd ()                                                                                                                                                                                                                                  |
| <b>Route</b>   | Edit, Find, Type, Specific Codes, select a code type, specify a code, Find or Find Next.<br>Edit, Replace, Type, Specific Codes, select a code type, specify a code, Replace or Replace All.                                                          |

---

**Example**

To search for a 12 point font size code, the command sequence is:

```
SpecificSearchStringBegin ()
FontSize (FontSize: 12p)
SpecificStringEnd ()
```

---

**See Also**

- ♦ SpecificDeleteStringBegin
- ♦ SpecificReplaceStringBegin
- ♦ SpecificSearchStringBegin

---

## SpreadsheetShowLinkCode

---

**Purpose**

SpreadsheetShowLinkCode specifies whether to display spreadsheet link codes.

---

**Syntax**

SpreadsheetShowLinkCode (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type: Specifies whether to display spreadsheet link codes.

0 No!  
1 Yes!

---

**Route**

Insert, Spreadsheet/Database, Options, Show Link Icons.

---

**Example**

To display spreadsheet link codes, the command is:

```
SpreadsheetShowLinkCode (State: Yes!)
```

---

**See Also**

- ♦ ImportOptionsDlg

---

## SpreadsheetUpdateOnRetrieve

---

**Purpose**

SpreadsheetUpdateOnRetrieve specifies whether to update spreadsheet links every time the document containing them is retrieved.

---

**Syntax**

SpreadsheetUpdateOnRetrieve (State: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to update spreadsheet links when retrieved with a document.

0 No!  
1 Yes!

---

**Route** Insert, Spreadsheet/Database, Options, Update on Retrieve.

---

**Example** To update all spreadsheet links when the document they are in is retrieved, the command is:

SpreadsheetUpdateOnRetrieve (State: Yes!)

---

**See Also** ♦ ImportOptionsDlg

---

## StatusBarShow

---

**Purpose** StatusBarShow specifies whether to display the Status Bar. If the parameter is omitted, this command acts as a toggle.

---

**Syntax** StatusBarShow (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display the Status Bar.

0 Off!  
1 On!

---

**Route** View, Status Bar.

---

**Example** To display the status bar, the command is:

StatusBarShow (State: On!)

---

# StyleCodes

---

**Purpose** StyleCodes edits style codes.  
IMPORTANT: This command must be used in conjunction with other commands, such as StyleEditBegin, StyleEditEnd, and SubstructureExit.

**Syntax** StyleCodes (State: *Numeric Equivalent* or *Enumerated Type*; Library: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to include codes that take effect when the style is turned off.

- 0 WithoutOffCodes!
- 1 WithOffCodes!

*Library* Numeric Equivalent or Enumerated Type: Specifies styles codes location.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Layout, Styles, select a style, Edit, Content.

---

**Example** To add a Times font code to a style named Heading, the command sequence is:

```
StyleEditBegin (Style: "Heading"; Library: CurrentDoc!)
StyleCodes (State: WithOffCodes!; Library: CurrentDoc!)
Font (Name: "Times")
SubstructureExit ()
StyleEditEnd (State: Save!)
```

---

**See Also** ♦ StyleEditBegin ♦ StyleEditEnd

---

# StyleCopy

---

**Purpose** StyleCopy copies a style between the current document, the personal library, and the shared library.

**Syntax** StyleCopy (Style: *Numeric Equivalent* or *Enumerated Type*; FromLibrary: *Numeric Equivalent* or *Enumerated Type*; ToLibrary: *Numeric Equivalent* or *Enumerated Type*; NewName: *Character Expression*)

---

## Parameters

### *Style*

Numeric Equivalent or Enumerated Type: Specifies the name of a style to copy. This parameter accepts a character expression to specify a user-defined style.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutInStyle!
- 46 OutlineStyle!
- 47 LegalOutInStyle!
- 48 BulletsOutInStyle!
- 49 HeadingsOutInStyle!
- 50 Legal\_2OutInStyle!
- 51 NumbersOutInStyle!
- 52 Level1Style!
- 53 Level2Style!
- 54 Level3Style!
- 55 Level4Style!
- 56 Level5Style!
- 57 Level6Style!
- 58 Level7Style!

- 59 Level8Style!
- 60 Legal1Style!
- 61 Legal2Style!
- 62 Legal3Style!
- 63 Legal4Style!
- 64 Legal5Style!
- 65 Legal6Style!
- 66 Legal7Style!
- 67 Legal8Style!
- 68 Heading1Style!
- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

*FromLibrary*

Numeric Equivalent or Enumerated Type: Specifies a style library source.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

*ToLibrary*

Numeric Equivalent or Enumerated Type: Specifies a destination style library.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

*NewName*

Character Expression: Specifies a new style name.

---

**Route**

Layout, Styles, select a style, Options, Copy, select a destination, OK, enter a style name (if applicable).

---

**Example**

To copy a style named Heading from the current document to the personal library, the command is:

```
StyleCopy (Style: "Heading"; FromLibrary: CurrentDoc!; ToLibrary: PersonalLibrary!; NewName: "MyStyle")
```

---

**See Also**

♦ StyleCreate ♦ StyleDelete

---

# StyleCreate

---

|                |                                                                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | StyleCreate creates a style.                                                                                                                                                                                                                  |
| <b>Syntax</b>  | StyleCreate (Name: <i>Character Expression</i> ; Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; CreateFlag: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

## Parameters

|                   |                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Name</i>       | Character Expression: Specifies a style name.                                                                                                                                                                 |
| <i>Type</i>       | Numeric Equivalent or Enumerated Type: Specifies a style type. <ol style="list-style-type: none"><li>1 OpenStyle!</li><li>2 CharacterStyle!</li><li>3 ParagraphStyle!</li></ol>                               |
| <i>CreateFlag</i> | Numeric Equivalent or Enumerated Type: Specifies a pattern to base a style on. <ol style="list-style-type: none"><li>1 CreateFromCurrent!</li><li>2 CreateBulletStyle!</li><li>3 CreateNumberStyle!</li></ol> |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies where to save a style. <ol style="list-style-type: none"><li>0 CurrentDoc!</li><li>1 PersonalLibrary!</li><li>2 SharedLibrary!</li></ol>                     |

---

|              |                                                                  |
|--------------|------------------------------------------------------------------|
| <b>Route</b> | Layout, Styles, Create, enter a style name, select a style type. |
|--------------|------------------------------------------------------------------|

---

|                |                                                                                                                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To create a paragraph style named Heading and save it in the Personal Library, the command is:<br><br>StyleCreate (Name: "Heading"; Type: ParagraphStyle!; CreateFlag: CreateFromCurrent!; Library: PersonalLibrary!) |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                 |
|-----------------|---------------------------------|
| <b>See Also</b> | ♦ StyleEditBegin ♦ StyleEditEnd |
|-----------------|---------------------------------|

---

# StyleDelete

---

|                |                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | StyleDelete deletes a style.                                                                                                                                                                          |
| <b>Syntax</b>  | StyleDelete (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Codes: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |



---

## Parameters

*Style*

Numeric Equivalent or Enumerated Type: Specifies the name of a style to delete. This parameter accepts a character expression to specify a user-defined style.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TextBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!
- 39 HeaderAStyle!
- 40 HeaderBStyle!
- 41 FooterAStyle!
- 42 FooterBStyle!
- 43 WatermarkAStyle!
- 44 WatermarkBStyle!
- 45 ParagraphOutlnStyle!
- 46 OutlineStyle!
- 47 LegalOutlnStyle!
- 48 BulletsOutlnStyle!
- 49 HeadingsOutlnStyle!
- 50 Legal\_2OutlnStyle!
- 51 NumbersOutlnStyle!
- 52 Level1Style!
- 53 Level2Style!
- 54 Level3Style!
- 55 Level4Style!
- 56 Level5Style!
- 57 Level6Style!
- 58 Level7Style!

- 59 Level8Style!
- 60 Legal1Style!
- 61 Legal2Style!
- 62 Legal3Style!
- 63 Legal4Style!
- 64 Legal5Style!
- 65 Legal6Style!
- 66 Legal7Style!
- 67 Legal8Style!
- 68 Heading1Style!
- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

*Codes* Numeric Equivalent or Enumerated Type: Specifies whether to delete document style codes with the style.

- 4 IncludingCodes!
- 5 LeavingCodes!

*Library* Numeric Equivalent or Enumerated Type: Specifies the library containing a style to delete.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

**Route** Layout, Styles, select a style, Options, Delete, select Include Codes or Leave Codes.

---

**Example** To delete a style named Heading from the default personal library, leaving the codes that are part of the style in the document, the command is:

StyleDelete (Style: "Heading"; Codes: LeavingCodes!; Library: PersonalLibrary!)

---

**See Also** ♦ StyleCopy ♦ StyleCreate

---

## StyleDescription

---

**Purpose** StyleDescription changes a style description.

IMPORTANT: This command must be used in conjunction with other commands, such as StyleEditBegin and StyleEditEnd.

|                    |                                                                                                                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>      | StyleDescription (Description: <i>Character Expression</i> )                                                                                                                                                                                                    |
| <b>Parameters</b>  |                                                                                                                                                                                                                                                                 |
| <i>Description</i> | Character Expression: Specifies a style description.                                                                                                                                                                                                            |
| <b>Route</b>       | Layout, Style, select a style, Edit, Description, enter a description.                                                                                                                                                                                          |
| <b>Example</b>     | To change the description of a style named Heading in the current document, the command sequence is:<br><br>StyleEditBegin (Style: "Heading"; Library: CurrentDoc!)<br>StyleDescription (Description: "1st level heading style")<br>StyleEditEnd (State: Save!) |
| <b>See Also</b>    | ♦ StyleCreate ♦ StyleEditBegin ♦ StyleEditEnd                                                                                                                                                                                                                   |

---

## StyleEditBegin

|                |                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | StyleEditBegin marks the start of changes to a style.<br><br>IMPORTANT: Any changes must appear between StyleEditBegin and StyleEditEnd.    |
| <b>Syntax</b>  | StyleEditBegin (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Library: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

### Parameters

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i> | Numeric Equivalent or Enumerated Type: Specifies the name of a style to edit. This parameter accepts a character expression to specify a user-defined style.                                                                                                                                                                                                                                                                                                                                                            |
|              | <ul style="list-style-type: none"> <li>0 NormalStyle!</li> <li>16 FootnoteNumberInDocumentStyle!</li> <li>17 EndnoteNumberInDocumentStyle!</li> <li>18 FigureNumberStyle!</li> <li>19 TableBoxNumberStyle!</li> <li>20 TextBoxNumberStyle!</li> <li>21 UserBoxNumberStyle!</li> <li>22 EquationNumberStyle!</li> <li>23 HypertextStyle!</li> <li>24 ToC1Style!</li> <li>25 ToC2Style!</li> <li>26 ToC3Style!</li> <li>27 ToC4Style!</li> <li>28 ToC5Style!</li> <li>29 Index1Style!</li> <li>30 Index2Style!</li> </ul> |

31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

## Library

Numeric Equivalent or Enumerated Type: Specifies a style library to edit.

- 0 CurrentDoc!
- 1 PersonalLibrary!
- 2 SharedLibrary!

---

|                 |                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Styles, select a style, Edit.                                                                                                                |
| <b>Example</b>  | To begin editing a style named Heading saved in the current document, the command is:<br><br>StyleEditBegin (Style: "Heading"; Library: CurrentDoc!) |
| <b>See Also</b> | ♦ StyleEditEnd                                                                                                                                       |

---

## StyleEditEnd

---

|                   |                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | StyleEditEnd marks the end of changes to a style and specifies whether to save the changes.<br><br>IMPORTANT: This command must be used in conjunction with StyleEditBegin. |
| <b>Syntax</b>     | StyleEditEnd (State: <i>Numeric Equivalent or Enumerated Type</i> )                                                                                                         |
| <b>Parameters</b> |                                                                                                                                                                             |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to save changes made to a style.<br><br>0 Cancel!<br>1 Save!                                                       |
| <b>Route</b>      | Layout, Styles, select a style, Edit, make editing changes.                                                                                                                 |
| <b>Example</b>    | To end style editing and save changes, the command is:<br><br>StyleEditEnd (State: Save!)                                                                                   |
| <b>See Also</b>   | ♦ StyleEditBegin                                                                                                                                                            |

---

---

# StyleEnterKeySetting

---

|                   |                                                                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | StyleEnterKeySetting specifies the effect of pressing Enter when a style is on.<br>IMPORTANT: This command must be used in conjunction with other commands, such as StyleEditBegin and StyleEditEnd. |
| <b>Syntax</b>     | StyleEnterKeySetting (Action: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                  |
| <b>Parameters</b> |                                                                                                                                                                                                      |
| <i>Action</i>     | Numeric Equivalent or Enumerated Type: Specifies the effect of pressing Enter.<br>0 HardReturn!<br>4 StyleOff!<br>12 StyleOffOn!                                                                     |
| <b>Route</b>      | Layout, Styles, select a style, Edit, Enter Key will Chain to, select an option.                                                                                                                     |
| <b>Example</b>    | To specify that Enter turns off a style named Heading, the command sequence is:<br><br>StyleEditBegin (Style: "Heading")<br>StyleEnterKeySetting (Action: StyleOff!)<br>StyleEditEnd (State: Save!)  |
| <b>See Also</b>   | ♦ StyleEditBegin ♦ StyleEditEnd ♦ StyleLinkTo                                                                                                                                                        |

---

# StyleLinkTo

---

|                   |                                                                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | StyleLinkTo specifies a style to turn on when Enter is pressed.<br>IMPORTANT: This command must be used in conjunction with other commands, such as StyleEditBegin and StyleEditEnd.                                                                                                                                         |
| <b>Syntax</b>     | StyleLinkTo (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                    |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                              |
| <i>Style</i>      | Numeric Equivalent or Enumerated Type: Specifies the name of a style to link to another style. This parameter accepts a character expression to specify a user-defined style.<br>0 NormalStyle!<br>16 FootnoteNumberInDocumentStyle!<br>17 EndnoteNumberInDocumentStyle!<br>18 FigureNumberStyle!<br>19 TableBoxNumberStyle! |

20 TextBoxNumberStyle!  
21 UserBoxNumberStyle!  
22 EquationNumberStyle!  
23 HypertextStyle!  
24 ToC1Style!  
25 ToC2Style!  
26 ToC3Style!  
27 ToC4Style!  
28 ToC5Style!  
29 Index1Style!  
30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!

68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

---

|                 |                                                                                                                                                                                                |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Layout, Styles, select a style, Edit, Enter Key will Chain to, specify a style.                                                                                                                |
| <b>Example</b>  | To link a style named Body Text to a style named Heading, the command sequence is:<br><br>StyleEditBegin (Style: "Heading")<br>StyleLinkTo (Style: "Body Text")<br>StyleEditEnd (State: Save!) |
| <b>See Also</b> | ♦ StyleEditBegin ♦ StyleEditEnd ♦ StyleEnterKeySetting                                                                                                                                         |

---

## StyleOff

---

|                |                                                                                                                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | StyleOff turns off a character style.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.<br><br>IMPORTANT: This command must be used in conjunction with other commands, such as StyleOn. |
| <b>Syntax</b>  | StyleOff (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                         |

---

### Parameters

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Style</i> | Numeric Equivalent or Enumerated Type: Specifies the name of a style to turn off. This parameter accepts a character expression to specify a user-defined style.<br><br>0 NormalStyle!<br>16 FootnoteNumberInDocumentStyle!<br>17 EndnoteNumberInDocumentStyle!<br>18 FigureNumberStyle!<br>19 TextBoxNumberStyle!<br>20 TextBoxNumberStyle!<br>21 UserBoxNumberStyle!<br>22 EquationNumberStyle!<br>23 HypertextStyle!<br>24 ToC1Style! |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



25 ToC2Style!  
26 ToC3Style!  
27 ToC4Style!  
28 ToC5Style!  
29 Index1Style!  
30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!

- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

---

**Example**

To turn off a style named Description, the command is:

StyleOff (Style: "Description")

---

## StyleOn

---

**Purpose**

StyleOn turns on a system style.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro. For a recordable command with similar functionality, use StyleSystemOn.

---

**Syntax**

StyleOn (Style: *Numeric Equivalent* or *Enumerated Type* or *Character Expression*;  
Action: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Style*

**Numeric Equivalent or Enumerated Type:** Specifies a style. Select a numeric equivalent or enumerated type to specify a pre-defined style, or use a character expression to specify a user-created style.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TextBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 TOC1Style!
- 25 TOC2Style!
- 26 TOC3Style!
- 27 TOC4Style!
- 28 TOC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!

37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to skip empty paragraphs.

1 SkipEmptyParagraphs!

---

**Example** To turn on the Heading 1 system style and skip empty paragraphs, the command is:

StyleOn (Style: Heading1Style! Action: SkipEmptyParagraphs!)

---

**See Also**

♦ StyleSystemOn ♦ StyleOff

---

## StyleRename

**Purpose**

StyleRename renames a style.

IMPORTANT: This command must be used in conjunction with other commands, such StyleEditBegin and StyleEditEnd.

**Syntax**

StyleRename (Name: *Character Expression*)

**Parameters***Name*

Character Expression: Specifies a new style name.

**Route**

Layout, Styles, select a style, Edit, Style Name, enter a name.

**Example**

To rename a style from Heading to Level1, the command sequence is:

```
StyleEditBegin (Style: "Heading")
StyleRename (Name: "Level1")
StyleEditEnd (State: Save!)
```

**See Also**

♦ StyleEditBegin ♦ StyleEditEnd

---

## StyleRetrieve

**Purpose**

StyleRetrieve retrieves the styles in a specified style file.

**Syntax**

StyleRetrieve (File: *Character Expression*; Styles: *Numeric Equivalent* or *Enumerated Type*; Library: *Numeric Equivalent* or *Enumerated Type*)

**Parameters***File*

Character Expression: Specifies the name of a style file to retrieve.

*Styles*

Numeric Equivalent or Enumerated Type: Specifies the type of style to retrieve.

- 1 UserStyles!
- 2 SystemStyles!
- 3 AllStyles!

|                 |                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Library</i>  | Numeric Equivalent or Enumerated Type: Specifies a destination library for the retrieved styles.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!          |
| <b>Route</b>    | Layout, Styles, Options, Retrieve, enter the name of a style file.                                                                                                   |
| <b>Example</b>  | To retrieve a style file named SCRIPT.STY into the current document, the command is:<br>StyleRetrieve (File: "SCRIPT.STY"; Styles: AllStyles!; Library: CurrentDoc!) |
| <b>See Also</b> | ♦ StyleSave                                                                                                                                                          |

## StyleSave

|                   |                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | StyleSave saves styles from a library to a style file.                                                                                                                                |
| <b>Syntax</b>     | StyleSave (Filename: <i>Character Expression</i> ; StyleType: <i>Numeric Equivalent or Enumerated Type</i> ; Library: <i>Numeric Equivalent or Enumerated Type</i> )                  |
| <b>Parameters</b> |                                                                                                                                                                                       |
| <i>Filename</i>   | Character Expression: Specifies the name of a style file to save.                                                                                                                     |
| <i>StyleType</i>  | Numeric Equivalent or Enumerated Type: Specifies style types to save.<br>1 UserStyles!<br>2 SystemStyles!<br>3 AllStyles!                                                             |
| <i>Library</i>    | Numeric Equivalent or Enumerated Type: Specifies the source library of styles to save.<br>0 CurrentDoc!<br>1 PersonalLibrary!<br>2 SharedLibrary!                                     |
| <b>Route</b>      | Layout, Styles, Options, Save As, enter a style file name.                                                                                                                            |
| <b>Example</b>    | To save the styles in the Personal Library in a style file named SCRIPT.STY, the command is:<br>StyleSave (Filename: "SCRIPT.STY"; StyleType: UserStyles!; Library: PersonalLibrary!) |

---

**See Also**

♦ StyleRetrieve

---

## StylesListDlg

---

**Purpose** StylesListDlg displays the Style List dialog box.

---

**Syntax** StylesListDlg ()

---

**Route** Layout, Styles.

---

## StyleSystemOn

---

**Purpose** StyleSystemOn turns on a system style or user-defined style.

---

**Syntax** StyleSystemOn (Style: *Numeric Equivalent* or *Enumerated Type* or *Character Expression*; Action: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Style* Numeric Equivalent or Enumerated Type: Specifies the name of a system style to turn on.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!

36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to skip empty paragraphs.

1 SkipEmptyParagraphs!

---

**Route** Layout, Styles, select a system style, Apply.

---

**Example** To turn on the Heading 1 system style, the command is:  
`StyleSystemOn (Style: Heading1Style!; Action: SkipEmptyParagraphs!)`

---

---

## StyleType

---

**Purpose** StyleType specifies a style type.  
IMPORTANT: This command must be used in conjunction with other commands, such as StyleEditBegin and StyleEditEnd.

---

**Syntax** StyleType (Type: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Type* Numeric Equivalent or Enumerated Type: Specifies a style type.

- 1 OpenStyle!
- 2 CharacterStyle!
- 3 ParagraphStyle!

---

**Route** Layout, Styles, select a style, Edit, Type, select a type.

---

**Example** To specify a paragraph style type for a style named Heading, the command sequence is:  
`StyleEditBegin (Style: "Heading")  
StyleType (Type: ParagraphStyle!)  
StyleEditEnd (State: Save!)`

---

**See Also** ♦ StyleEditBegin ♦ StyleEditEnd

---

---

## SubdividePage

---

**Purpose** SubdividePage divides a physical page into two or more logical pages.

---

**Syntax** SubdividePage (Columns: *Numeric Expression*; Rows: *Numeric Expression*)

---

**Parameters**



|                |                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Columns</i> | Numeric Expression: Specifies the number of logical page columns on a physical page.                                                        |
| <i>Rows</i>    | Numeric Expression: Specifies the number of logical page rows on a physical page.                                                           |
| <b>Route</b>   | Layout, Page, Subdivide Page, enter the number of columns and rows.                                                                         |
| <b>Example</b> | To subdivide a physical page into four logical pages (two columns and two rows), the command is:<br><br>SubdividePage (Columns: 2; Rows: 2) |

---

## SubdividePageDlg

|                |                                                          |
|----------------|----------------------------------------------------------|
| <b>Purpose</b> | SubdividePageDlg displays the Subdivide Page dialog box. |
| <b>Syntax</b>  | SubdividePageDlg ()                                      |
| <b>Route</b>   | Layout, Page, Subdivide Page.                            |

---

## SubdividePageOff

|                 |                                                              |
|-----------------|--------------------------------------------------------------|
| <b>Purpose</b>  | SubdividePageOff turns off the Layout option Subdivide Page. |
| <b>Syntax</b>   | SubdividePageOff ()                                          |
| <b>Route</b>    | Layout, Page, Subdivide Page, Off.                           |
| <b>See Also</b> | ♦ SubdividePage                                              |

---

## SubdocInclude

|                |                                                                              |
|----------------|------------------------------------------------------------------------------|
| <b>Purpose</b> | SubdocInclude inserts a document into the current document as a subdocument. |
| <b>Syntax</b>  | SubdocInclude (DocumentName: <i>Character Expression</i> )                   |

---

## Parameters

*DocumentName* Character Expression: Specifies the name of a file to insert as a subdocument.

---

**Route** File, Master Document, Subdocument, enter a filename, Include.

---

**Example** To insert a document named CHAPTER3 as a subdocument, the command is:

SubdocInclude (DocumentName: "CHAPTER3")

---

---

## SubstructureExit

**Purpose** SubstructureExit closes the current substructure.

---

**Syntax** SubstructureExit (Action: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to open the previous or the next substructure of the same type.

1 Previous!  
15 Next!

---

**Route** From a feature bar, choose Next, Previous, or Close.

---

**Example** To exit the current substructure and enter the next, the command is:

SubstructureExit (Action: Next!)

---

---

## Suppress

**Purpose** Suppress turns off specified page formatting item(s) for the current page.

---

**Syntax** Suppress (Item: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Item* Numeric Equivalent or Enumerated Type: Specifies page format items to suppress.

0 PageNumbering!  
1 PageNumberBottomCenter!  
2 HeaderA!  
3 HeaderB!  
4 FooterA!

- 5 FooterB!
- 6 WatermarkA!
- 7 WatermarkB!

---

|                |                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Layout, Page, Suppress, select items.                                                                                           |
| <b>Example</b> | To turn off Page Numbering and Header A on the current page, the command is:<br>Suppress (Item: PageNumbering!; Item: HeaderA!) |

---

## SwitchDoc

---

|                   |                                                                   |
|-------------------|-------------------------------------------------------------------|
| <b>Purpose</b>    | SwitchDoc specifies a document to switch to.                      |
| <b>Syntax</b>     | SwitchDoc (DocNum: <i>Numeric Expression</i> )                    |
| <b>Parameters</b> |                                                                   |
| <i>DocNum</i>     | Numeric Expression                                                |
| <b>Example</b>    | To switch to document 3, the command is:<br>SwitchDoc (DocNum: 3) |

---

## Tab

---

|                 |                    |
|-----------------|--------------------|
| <b>Purpose</b>  | Tab inserts a tab. |
| <b>Syntax</b>   | Tab ()             |
| <b>Route</b>    | Press Tab.         |
| <b>See Also</b> | ♦ TabKey ♦ TabSet  |

---

## TabAdd

---

|                |                                                                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TabAdd adds one or more tab sets to the current setting. If the parameter SpaceBetweenTabs is included, all tabs following an added tab will be reset to the new type and spacing. |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** TabAdd (Position: *Numeric Expression*; Type: *Numeric Equivalent* or *Enumerated Type*; SpaceBetweenTabs: *Numeric Expression*)

---

**Parameters**

*Position* Numeric Expression: Specifies a tab position relative to the left edge of the page.

*Type* Numeric Equivalent or Enumerated Type: Specifies a tab type.

- 0 TabLeft!
- 1 TabCenter!
- 2 TabRight!
- 3 TabDecimal!
- 16 TabLeftDot!
- 17 TabCenterDot!
- 18 TabRightDot!
- 19 TabDecimalDot!

*SpaceBetweenTabs* Measurement Expression: Specifies the distance between tabs.

---

**Example** To add a center tab 3.2 inches from the left edge of the page, the command is:

```
TabAdd (Position: 3.2"; Type: TabCenter!; SpaceBetweenTabs: .5")
```

---

**See Also** ♦ Tab ♦ TabSet

---

---

## TabCenter

---

**Purpose** TabCenter inserts a Hard Centered Tab code at the insertion point.

**Syntax** TabCenter (Type: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Type* Numeric Equivalent or Enumerated Type: Specifies whether the Hard Centered Tab code includes a dot leader.

- 0 Normal!
  - 1 DotLeader!
- 

**Route** Layout, Line, Other Codes, select Center under Hard Tab Codes or Center under Hard Tab Codes with Dot Leaders, Insert.

---

---

|                 |                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Example</b>  | To insert a Hard Centered Tab code with a dot leader, the command is:<br><code>TabCenter (Type: DotLeader!)</code> |
| <b>See Also</b> | ♦ TabDecimal ♦ TabLeft ♦ TabRight                                                                                  |

---

## TabDecimal

---

|                   |                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TabDecimal inserts a decimal align tab.                                                                                         |
| <b>Syntax</b>     | TabDecimal (Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                         |
| <b>Parameters</b> |                                                                                                                                 |
| <i>Type</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to include a dot leader.<br>0 Normal!<br>1 DotLeader!                  |
| <b>Route</b>      | Layout, Line, Other Codes, select Decimal under Hard Tab Codes options or Center under Hard Tab Codes with Dot Leaders options. |
| <b>Example</b>    | To insert a decimal tab with a dot leader, the command is:<br><code>TabDecimal (Type: DotLeader!)</code>                        |
| <b>See Also</b>   | ♦ TabCenter ♦ TabLeft ♦ TabRight                                                                                                |

---

## TabDelete

---

|                   |                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TabDelete deletes one tab setting without affecting any other tab settings.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>     | TabDelete (Position: <i>Measurement Expression</i> )                                                                                                                            |
| <b>Parameters</b> |                                                                                                                                                                                 |
| <i>Position</i>   | Measurement Expression: Specifies a tab setting position to remove.                                                                                                             |

---

---

**Example** To remove a tab setting at 3 inches, the command is:  
TabDelete (Position: 3")

---

**See Also** ♦ Tab ♦ TabAdd ♦ TabSet

---

---

## TabInsert

---

**Purpose** TabInsert inserts a tab at the insertion point in the current document or table.

---

**Syntax** TabInsert ()

---

**Route** Press Ctrl+Tab.

---

---

## TabKey

---

**Purpose** TabKey performs the functions of Tab.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** TabKey ()

---

**See Also** ♦ Tab

---

---

## TableAdjustBegin

---

**Purpose** TableAdjustBegin is the opening command of the TableAdjustBegin-TableAdjustEnd statement. These commands are entered when adjusting or positioning the table from the Ruler Bar.  
IMPORTANT: Between these commands may come other commands, such as TableAdjustColumnWidth and TableAdjustPosition.

---

**Syntax** TableAdjustBegin ()

---

**Route** With the insertion point in a table, View, Ruler Bar, drag column or margin markers to a new position.

---

---

## TableAdjustColumnWidth

---

**Purpose** TableAdjustColumnWidth specifies a column width in WordPerfect units (100 units equals one inch). This command is entered when adjusting columns from the ruler bar.

IMPORTANT: This command must be used with the TableAdjustBegin and the TableAdjustEnd commands.

---

**Syntax** TableAdjustColumnWidth (Column: *Numeric Expression*; Width: *Numeric Expression*)

---

### Parameters

*Column* Numeric Expression: Specifies a column number. The farthest column to the left is 0.

*Width* Numeric Expression: Specifies column width.

---

**Route** With the insertion point in a table, View, Ruler Bar, drag a column marker to a new position.

---

**Example** To specify a column width of 2 inches, the command is:

TableAdjustColumnWidth (Column: 1 Width: 200)

---

---

## TableAdjustEnd

---

**Purpose** TableAdjustEnd is the closing command of the TableAdjustBegin-TableAdjustEnd statement. These commands are entered when adjusting or positioning the table from the Ruler Bar.

IMPORTANT: Between these commands may come other commands, such as TableAdjustColumnWidth and TableAdjustPosition.

---

**Syntax** TableAdjustEnd ()

---

**Route** With the insertion point in a table, View, Ruler Bar, drag column or margin markers to a new position.

---

---

## TableAdjustPosition

---

|                   |                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableAdjustPosition specifies in WordPerfect units the horizontal position of a table relative to the left margin. This command is entered when adjusting the table position from the Ruler Bar. |
| <b>Syntax</b>     | TableAdjustPosition (Position: <i>Numeric Expression</i> )                                                                                                                                       |
| <b>Parameters</b> |                                                                                                                                                                                                  |
| <i>Position</i>   | Numeric Expression: Specifies table position relative to the left margin.                                                                                                                        |
| <b>Route</b>      | With the insertion point in a table, View, Ruler Bar, press Ctrl and drag margin markers to move the table.                                                                                      |
| <b>Example</b>    | To position a table one inch to the right of the left margin, the command is:<br><code>TableAdjustPosition (Position: 100)</code>                                                                |

---

## TableAppendRow

---

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| <b>Purpose</b>  | TableAppendRow inserts a new row below the current row. |
| <b>Syntax</b>   | TableAppendRow ()                                       |
| <b>Route</b>    | Press Alt+Shift+Insert.                                 |
| <b>See Also</b> | ♦ TableInsertRow                                        |

---

## TableAttributeOff

---

|                   |                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableAttributeOff turns off one or more character attributes in a table.                                                                                                                           |
| <b>Syntax</b>     | TableAttributeOff (Attribute: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                |
| <b>Parameters</b> |                                                                                                                                                                                                    |
| <i>Attribute</i>  | Numeric Equivalent or Enumerated Type: Specifies an attribute to turn off.<br><ul style="list-style-type: none"><li>0 ExtraLarge!</li><li>1 VeryLarge!</li><li>2 Large!</li><li>3 Small!</li></ul> |



- 4 Fine!
- 5 Superscript!
- 6 Subscript!
- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!
- 16 Every!

|                 |                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Format, Table, deselect options.                                                             |
| <b>Example</b>  | To turn off all attributes in a table, the command is:<br><br>TableAttributeOff (Attribute: Every!) |
| <b>See Also</b> | ♦ TableAttributeOn                                                                                  |

---

## TableAttributeOn

|                   |                                                                                    |
|-------------------|------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableAttributeOn turns on one or more character attributes in a table.             |
| <b>Syntax</b>     | TableAttributeOn (Attribute: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                    |

|                  |                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Attribute</i> | Numeric Equivalent or Enumerated Type: Specifies an attribute to turn on.                                                                                                                                                                                                                                                                      |
|                  | <ul style="list-style-type: none"> <li>0 ExtraLarge!</li> <li>1 VeryLarge!</li> <li>2 Large!</li> <li>3 Small!</li> <li>4 Fine!</li> <li>5 Superscript!</li> <li>6 Subscript!</li> <li>7 Outline!</li> <li>8 Italics!</li> <li>9 Shadow!</li> <li>10 Redline!</li> <li>11 DoubleUnderline!</li> <li>12 Bold!</li> <li>13 Strikeout!</li> </ul> |

- 14 Underline!
- 15 SmallCaps!

---

**Route** Table, Format, Table, select options.

---

**Example** To turn on Bold and Underline in a table, the command is:

TableAttributeOn (Attribute: Bold!; Attribute: Underline!)

---

**See Also** ♦ TableAttributeOff

---

---

## TableAutoCalc

---

**Purpose** TableAutoCalc specifies automatic calculation for tables.

---

**Syntax** TableAutoCalc (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies automatic calculation for tables.

- 0 Off!
- 1 AutoTbl!
- 2 AutoDoc!

---

**Route** Table, Calculate, select an option.

---

**Example** To automatically calculate a document, the command is:

TableAutoCalc (State: AutoDoc!)

---

**See Also** ♦ TableCalculate ♦ TableCalculateAll ♦ TableCalculateDlg

---

---

## TableAutoFill

---

**Purpose** TableAutoFill uses TableDataFill to increment values across a row or down a column. IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** TableAutoFill ()

---

---

## TableAutoFormulaEntry

---

**Purpose** TableAutoFormulaEntry specifies whether formulas can be entered in tables at the document level or only in table editing mode.

**Syntax** TableAutoFormulaEntry (State: *Numeric Equivalent* or *Enumerated Type*)

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether formula entry is on.

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

**Route** Table, Cell Formula Entry.

**Example** To allow formula entry at the document level, the command is:

TableAutoFormulaEntry (State: On!)

**See Also** ♦ TableFormula ♦ TableTextToFormula

---

## TableBlockOff

---

**Purpose** TableBlockOff turns Select off in a table.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** TableBlockOff ()

---

## TableBlockOn

---

**Purpose** TableBlockOn turns Select on in a table.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** TableBlockOn ()

---

# TableBorder

---

**Purpose** TableBorder specifies a border type for a table.  
IMPORTANT: This command must be used in conjunction with other commands, such as TableBorderEditBegin and TableBorderEditEnd.

---

**Syntax** TableBorder (Style: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies a border type.

- 0 SpacingOnly!
- 1 SingleBorder!
- 2 DoubleBorder!
- 3 DashedBorder!
- 4 DottedBorder!
- 5 ThickBorder!
- 6 ExtraThickBorder!
- 7 ThinThickBorder!
- 8 ThickThinBorder!
- 9 ThickTopBottomBorder!
- 10 ButtonBorder!
- 11 ColumnBorderBetween!
- 12 ColumnBorderAll!
- 127 NoBorder!

---

**Route** Table, Lines/Fill, Table, Border Lines.

---

**Example** To specify an extra thick table border, the command sequence is:

```
TableBorderEditBegin ()
TableBorder (Style: ExtraThickBorder!)
TableBorderEditEnd (State: Save!)
```

---

**See Also** ♦ TableBorderEditBegin ♦ TableBorderEditEnd

---

---

# TableBorderEditBegin

---

**Purpose** TableBorderEditBegin initiates table border editing.  
IMPORTANT: Border commands must appear between this command and TableBorderEditEnd.

|                 |                                         |
|-----------------|-----------------------------------------|
| <b>Syntax</b>   | TableBorderEditBegin ()                 |
| <b>Route</b>    | Table, Lines/Fill, Table, Border Lines. |
| <b>See Also</b> | ♦ TableBorderEditEnd                    |

---

## TableBorderEditEnd

|                |                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableBorderEditEnd marks the end of table border editing.<br>IMPORTANT: This command must be used in conjunction with TableBorderEditBegin. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|

|               |                                                                                  |
|---------------|----------------------------------------------------------------------------------|
| <b>Syntax</b> | TableBorderEditEnd (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|----------------------------------------------------------------------------------|

|                   |                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------|
| <b>Parameters</b> |                                                                                                   |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to save changes.<br>0 Cancel!<br>1 Save! |

|              |                                         |
|--------------|-----------------------------------------|
| <b>Route</b> | Table, Lines/Fill, Table, Border Lines. |
|--------------|-----------------------------------------|

|                |                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify no table border, the command sequence is:<br>TableBorderEditBegin ()<br>TableBorder (State: NoBorder!)<br>TableBorderEditEnd (State: Save!) |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|

|                 |                        |
|-----------------|------------------------|
| <b>See Also</b> | ♦ TableBorderEditBegin |
|-----------------|------------------------|

---

## TableCalculate

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | TableCalculate calculates formula values in the current table. |
|----------------|----------------------------------------------------------------|

|               |                   |
|---------------|-------------------|
| <b>Syntax</b> | TableCalculate () |
|---------------|-------------------|

|              |                               |
|--------------|-------------------------------|
| <b>Route</b> | Table, Calculate, Calc Table. |
|--------------|-------------------------------|

---

**See Also**

♦ TableCalculateAll ♦ TableFormula

---

## TableCalculateAll

---

**Purpose**

TableCalculateAll calculates formula values for all tables and floating cells in the current document.

**Syntax**

TableCalculateAll ()

**Route**

Table, Calculate, Calc Document.

**See Also**

♦ TableCalculate

---

## TableCalculateDlg

---

**Purpose**

TableCalculateDlg displays the Calculate dialog box.

**Syntax**

TableCalculateDlg ()

**Route**

Table, Calculate.

---

## TableCellAttributeOff

---

**Purpose**

TableCellAttributeOff turns off one or more character attributes in a table cell.

**Syntax**

TableCellAttributeOff ({Attribute: *Numeric Equivalent* or *Enumerated Type*})

**Parameters***Attribute*

Numeric Equivalent or Enumerated Type: Specifies an attribute to turn off. Enclose multiple attributes in braces ({}), separated by semicolons.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 Superscript!
- 6 Subscript!

- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!
- 16 Every!

---

**Route** Table, Format, Cell, deselect options.

---

**Example** To turn off Bold and Underline in a table cell, the command is:  
`TableCellAttributeOff ({Attribute: Bold!: Attribute: Underline!})`

---

**See Also** ♦ TableCellAttributeOn

---

## TableCellAttributeOn

---

**Purpose** TableCellAttributeOn turns on one or more character attributes in a table cell.

---

**Syntax** TableCellAttributeOn ({Attribute: *Numeric Equivalent* or *Enumerated Type*})

---

### Parameters

*Attribute* Numeric Equivalent or Enumerated Type: Specifies an attribute to turn on. Enclose multiple attributes in braces ({}), separated by semicolons.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 Superscript!
- 6 Subscript!
- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!

|                 |                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Format, Cell, select options.                                                                                                        |
| <b>Example</b>  | To turn on Italics and Large in a table cell, the command is:<br><code>TableCellAttribute ({Attribute: Italics!; Attribute: Large!})</code> |
| <b>See Also</b> | ♦ TableCellAttributeOff                                                                                                                     |

---

## TableCellCopy

|                   |                                                                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableCellCopy specifies where in a table to copy the contents of a cell.                                                                                                                                                                     |
| <b>Syntax</b>     | TableCellCopy (CopyType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ; Count: <i>Numeric Expression</i> )                                                                                                                            |
| <b>Parameters</b> |                                                                                                                                                                                                                                              |
| <i>CopyType</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to copy the contents of the current cell down, to the right, or to a specific cell.<br><ul style="list-style-type: none"> <li>0 Down!</li> <li>3 Right!</li> <li>4 Cell!</li> </ul> |
| <i>Count</i>      | Numeric Expression: Specifies the number of cells to copy down or to the right. This parameter is used only when Down! or Right! is selected as the destination parameter.                                                                   |
| <b>Example</b>    | To copy the contents of a selected cell four times to the right, the command is:<br><code>TableCellCopy (CopyType: Right!; Count: 4)</code>                                                                                                  |
| <b>See Also</b>   | ♦ EditCopy ♦ EditPaste                                                                                                                                                                                                                       |

---

## TableCellFillColors

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableCellFillColors specifies foreground and background colors for a cell or blocked cells.                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Syntax</b>  | TableCellFillColors (ForeColorName: <i>Character Expression</i> ; ForeColorRed: <i>Numeric Expression</i> ; ForeColorGreen: <i>Numeric Expression</i> ; ForeColorBlue: <i>Numeric Expression</i> ; ForeColorShade: <i>Numeric Expression</i> ; BackColorName: <i>Character Expression</i> ; BackColorRed: <i>Numeric Expression</i> ; BackColorGreen: <i>Numeric Expression</i> ; BackColorBlue: <i>Numeric Expression</i> ; BackColorShade: <i>Numeric Expression</i> ) |



---

## Parameters

|                       |                                                                                                  |
|-----------------------|--------------------------------------------------------------------------------------------------|
| <i>ForeColorName</i>  | Character Expression: Specifies a predefined color name.                                         |
| <i>ForeColorRed</i>   | Numeric Expression: Specifies the amount of red in a custom color. Values range from 0 to 255.   |
| <i>ForeColorGreen</i> | Numeric Expression: Specifies the amount of green in a custom color. Values range from 0 to 255. |
| <i>ForeColorBlue</i>  | Numeric Expression: Specifies the amount of blue in a custom color. Values range from 0 to 255.  |
| <i>ForeColorShade</i> | Numeric Expression: Specifies the shading percentage in a custom color.                          |
| <i>BackColorName</i>  | Character Expression: Specifies a predefined color name.                                         |
| <i>BackColorRed</i>   | Numeric Expression: Specifies the amount of red in a custom color. Values range from 0 to 255.   |
| <i>BackColorGreen</i> | Numeric Expression: Specifies the amount of green in a custom color. Values range from 0 to 255. |
| <i>BackColorBlue</i>  | Numeric Expression: Specifies the amount of blue in a custom color. Values range from 0 to 255.  |
| <i>BackColorShade</i> | Numeric Expression: Specifies the shading percentage in a custom color.                          |

---

## Route

Table, Lines/Fill, Current Cell or Selection, select a fill style, foreground color, and background color.

---

## Example

To specify a cyan foreground and a magenta background, the command is:

```
TableCellFillColors (ForeColorName: "Magenta"; ForeColorRed: 255; ForeColorGreen: 0; ForeColorBlue: 255; ForeColorShade: 100; BackColorName: "Cyan"; BackColorRed: 0; BackColorGreen: 255; BackColorBlue: 255; BackColorShade: 100)
```

---

## See Also

- [TableCellFillStyle](#)

---

# TableCellFillStyle

---

## Purpose

TableCellFillStyle specifies a shading style for a cell or blocked cells.

---

## Syntax

TableCellFillStyle (FillPattern: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

|                    |                                                                                                                                                                                                                                    |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>FillPattern</i> | Numeric Equivalent or Enumerated Type: Specifies a fill style.<br>0 Fill10!<br>1 Fill20!<br>2 Fill30!<br>3 Fill40!<br>4 Fill50!<br>5 Fill60!<br>6 Fill70!<br>7 Fill80!<br>8 Fill90!<br>9 Fill100!<br>10 FillButton!<br>127 NoFill! |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|              |                                                                    |
|--------------|--------------------------------------------------------------------|
| <b>Route</b> | Table, Lines/Fill, Current Cell or Selection, select a fill style. |
|--------------|--------------------------------------------------------------------|

---

|                |                                                                                                     |
|----------------|-----------------------------------------------------------------------------------------------------|
| <b>Example</b> | To fill a table cell with 20% shading, the command is:<br>TableCellFillStyle (FillPattern: Fill20!) |
|----------------|-----------------------------------------------------------------------------------------------------|

---

|                 |                       |
|-----------------|-----------------------|
| <b>See Also</b> | ♦ TableCellFillColors |
|-----------------|-----------------------|

---

## TableCellIgnoreCalculation

---

|                |                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableCellIgnoreCalculation specifies one or more selected cells to ignore when table calculations are performed. |
|----------------|------------------------------------------------------------------------------------------------------------------|

---

|               |                                                                                   |
|---------------|-----------------------------------------------------------------------------------|
| <b>Syntax</b> | TableCellIgnoreCalculation (State: <i>Numeric Equivalent or Enumerated Type</i> ) |
|---------------|-----------------------------------------------------------------------------------|

---

**Parameters**

|              |                                                                                                                                             |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether to ignore one or more selected cells when calculating formulas.<br>0 No!<br>1 Yes! |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------|

---

|              |                                              |
|--------------|----------------------------------------------|
| <b>Route</b> | Table, Format, Ignore Cell When Calculating. |
|--------------|----------------------------------------------|

---

|                |                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To ignore the current cell when calculating, the command is:<br>TableCellIgnoreCalculation (State: Yes!) |
|----------------|----------------------------------------------------------------------------------------------------------|

---

**See Also**

♦ TableCalculate

---

## TableCellJustification

---

**Purpose**

TableCellJustification specifies the justification for a cell or blocked cells.

---

**Syntax**

TableCellJustification (Mode: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Mode*

Numeric Equivalent or Enumerated Type: Specifies the justification for a cell or selected cells.

- 0 Left!
- 1 Full!
- 2 Center!
- 3 Right!
- 4 FullAll!
- 5 DecAlign!

---

**Route**

Table, Format, Justification, select option.

---

**Example**

To specify right justification for a cell, the command is:

TableCellJustification (Mode: Right!)

---

**See Also**

♦ TableColumnJustification ♦ TableFormatJustification

---

## TableCellLine

---

**Purpose**

TableCellLine changes specified lines around a cell or selected cells, or between selected cells.

---

**Syntax**

TableCellLine (Position: *Numeric Equivalent* or *Enumerated Type*; Style: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Position*

Numeric Equivalent or Enumerated Type: Specifies a line or lines to change.

- 0 Left!
- 1 Top!
- 2 Inside!
- 3 Right!

- 4 Bottom!
- 5 Outside!

*Style*

Numeric Equivalent or Enumerated Type: Specifies a line style.

- 0 SingleLine!
- 1 DoubleLine!
- 2 DashedLine!
- 3 DottedLine!
- 4 ThickLine!
- 5 ExtraThickLine!
- 6 ThinThickLine!
- 7 ThickThinLine!
- 8 ButtonTopLeftLine!
- 9 ButtonBottomRightLine!
- 126 TableDefaultLine!
- 127 NoLine!

---

**Route**

Table, Lines/Fill, Current Cell or Selection, select options from the Line Styles group box.

---

**Example**

To create a thick line around one or more selected cells, the command is:

TableCellLine (Position: Outside!; Style: ThickLine!)

---

**See Also**

- ♦ TableCellLineColor

---

## TableCellLineColor

---

**Purpose**

TableCellLineColor specifies the line color of one or more selected cells.

---

**Syntax**

TableCellLineColor (ColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; Shade: *Numeric Expression*)

---

**Parameters**

*ColorName*

Character Expression: Specifies a predefined color name.

*Red*

Numeric Expression: Specifies the amount of red in a custom color. Values range from 0 to 255.

*Green*

Numeric Expression: Specifies the amount of green in a custom color. Values range from 0 to 255.

|                 |                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Blue</i>     | Numeric Expression: Specifies the amount of blue in a custom color. Values range from 0 to 255.                                            |
| <i>Shade</i>    | Numeric Expression: Specifies the shading percentage in a custom color.                                                                    |
| <b>Route</b>    | Table, Lines/Fill, Current Cell or Selection, select a color from the Line Color group box.                                                |
| <b>Example</b>  | To specify magenta cell lines, the command is:<br><br>TableCellLineColor (ColorName: "Magenta"; Red: 255; Green: 0; Blue: 255; Shade: 100) |
| <b>See Also</b> | ♦ TableCellLine                                                                                                                            |

---

## TableCellLock

|                   |                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableCellLock specifies whether to lock a cell or selected cells to prevent editing.                              |
| <b>Syntax</b>     | TableCellLock (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                       |
| <b>Parameters</b> |                                                                                                                   |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to lock a cell or selected cells.<br><br>0 No!<br>1 Yes! |
| <b>Route</b>      | Table, Format, Cell, Lock.                                                                                        |
| <b>Example</b>    | To lock a cell, the command is:<br><br>TableCellLock (State: Yes!)                                                |

---

## TableCellNumberAlignCurrency

|                |                                                                                            |
|----------------|--------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableCellNumberAlignCurrency aligns currency symbols in one or more selected cells.        |
| <b>Syntax</b>  | TableCellNumberAlignCurrency (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to align currency symbols.

- 0 No!
- 1 Yes!

---

**Route** Table, Number Type, Accounting or Currency, Custom, Align Currency Symbol.

---

**Example** To align currency symbols, the command is:

TableCellNumberAlignCurrency (State: Yes!)

---

**See Also** ♦ TableCellNumberSelectCurrency

---

## TableCellNumberDateFormat

---

**Purpose** TableCellNumberDateFormat specifies a date format for one or more selected cells.

---

**Syntax** TableCellNumberDateFormat (DateFormatNumber: *Numeric Expression*)

---

## Parameters

*DateFormatNumber* Numeric Expression: Specifies a date format.

- 1 July 5, 1959
- 2 7/5/59
- 3 Jul 5, 1959
- 4 5 July 1959
- 5 Sunday, July 5, 1959
- 6 12:00 pm
- 7 05Jul59
- 8 July 5, 1959 (12:00pm)
- 9 1959-07-05
- 10 7/5
- 11 Jul 5
- 12 12:00:00

---

**Route** Table, Number Type, Date/Time, Custom, select a Date/Time type.

---

**Example** To display the date in the format Sunday, July 5, 1959, the command is:

TableCellNumberDateFormat (DateFormatNumber: 5)

---

**See Also** ♦ TableCellNumberFormat

---

## TableCellNumberDecimalDigits

---

**Purpose** TableCellNumberDecimalDigits specifies how many digits appear after the decimal point.

IMPORTANT: Use with the Fixed2! option of TableCellNumberFormat.

---

**Syntax** TableCellNumberDecimalDigits (NumDigits: *Numeric Expression*)

---

### Parameters

*NumDigits* Numeric Expression: Specifies how many digits appear after the decimal point.

---

**Route** Table, Number Type, Cell, Fixed, Custom, specify the number of digits after the decimal point.

---

**Example** To specify three digits after the decimal point, the command sequence is:  
    TableCellNumberDecimalDigits (NumDigits: 3)

---

**See Also** ♦ TableCellNumberFormat

---

## TableCellNumberFormat

---

**Purpose** TableCellNumberFormat specifies a number format for one or more selected cells.

---

**Syntax** TableCellNumberFormat (FormatType: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*FormatType* Numeric Equivalent or Enumerated Type: Specifies a number format.

- 0 General!
- 1 Integer!
- 2 Fixed2!
- 3 Percent!
- 4 Currency!
- 5 Accounting!
- 6 Commas!
- 7 Scientific!
- 8 DateFormat!
- 9 TextOnly!

|                 |                                                                                                                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Cell, select a type.                                                                                                                                                                                                                                              |
| <b>Example</b>  | To specify currency as the number type, the command is:<br><code>TableCellNumberFormat (FormatType: Currency!)</code>                                                                                                                                                                 |
| <b>See Also</b> | <ul style="list-style-type: none"> <li>♦ <code>TableCellNumberAlignCurrency</code> ♦ <code>TableCellNumberDateFormat</code></li> <li>♦ <code>TableCellNumberDecimalDigits</code> ♦ <code>TableCellNumberRound</code></li> <li>♦ <code>TableCellNumberSelectCurrency</code></li> </ul> |

---

## TableCellNumberNegativeNumber

|                       |                                                                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>        | <code>TableCellNumberNegativeNumber</code> specifies how to display negative numbers in one or more selected cells.                                                                               |
| <b>Syntax</b>         | <code>TableCellNumberNegativeNumber</code> (NegDisplayType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                 |
| <b>Parameters</b>     |                                                                                                                                                                                                   |
| <i>NegDisplayType</i> | <p>Numeric Equivalent or Enumerated Type: Specifies a display style for negative numbers.</p> <ul style="list-style-type: none"> <li>0 Minus!</li> <li>1 Parentheses!</li> <li>2 CRDR!</li> </ul> |

|                |                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Table, Number Type, Cell, Custom, select a negative number option.                                                                      |
| <b>Example</b> | To display numbers using the Credit/Debit method, the command is:<br><code>TableCellNumberNegativeNumber (NegDisplayType: CRDR!)</code> |

---

## TableCellNumberRound

|                |                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <p><code>TableCellNumberRound</code> specifies a method for converting numbers entered as decimals to integers.</p> <p>IMPORTANT: Use with the Integer! option of <code>TableCellNumberFormat</code>.</p> |
| <b>Syntax</b>  | <code>TableCellNumberRound</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                           |



---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to round off numbers or truncate them when converting to integers.

- 0 No!
- 1 Yes!

---

**Route** Table, Number Type, Cell, Custom, Round for Calculation.

---

**Example** To round off numbers after the decimal point when converting them to integers, the command is:

TableCellNumberRound (State: Yes!)

---

**See Also** ♦ TableCellNumberFormat

---

## TableCellNumberSelectCurrency

---

**Purpose** TableCellNumberSelectCurrency specifies a currency type for the current cell or block of cells.

---

**Syntax** TableCellNumberSelectCurrency (IntlCurrency: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*IntlCurrency* Numeric Equivalent or Enumerated Type: Specifies a currency type.

- 1 Dollar!
- 2 Peso!
- 3 FrenchCanada!
- 4 Canada!
- 5 Brazil!
- 6 Denmark!
- 7 Peseta!
- 8 Croatia!
- 9 CzechRepublic!
- 10 SlovakRepublic!
- 11 Finland!
- 12 FinlandIntl!
- 13 France!
- 14 Germany!
- 15 Greece!
- 16 Hungary!
- 17 Iceland!
- 18 Italy!

- 19 Japan!
- 20 Netherlands!
- 21 Norway!
- 22 NorwayIntl!
- 23 Ruble!
- 24 RussianRuble!
- 25 SouthAfrica!
- 26 Sweden!
- 27 Turkey!
- 28 UnitedKingdom!
- 29 Switzerland!
- 30 Austria!
- 31 Belgium!
- 32 Ecuador!
- 33 ElSalvador!
- 34 Guatemala!
- 35 Honduras!
- 36 Paraguay!
- 37 Venezuela!
- 38 Poland!
- 39 Slovenia!

---

**Route** Table, Number Type, Cell, Accounting or Currency, Custom, select a symbol.

---

**Example** To specify pesos as the currency type, the command is:  
TableCellNumberSelectCurrency (IntlCurrency: Peso!)

---

**See Also** ♦ TableCellNumberAlignCurrency ♦ TableCellNumberFormat

---

## TableCellNumberUseCommas

---

**Purpose** TableCellNumberUseCommas displays commas in numbers with more than three digits to the left of the decimal point.

---

**Syntax** TableCellNumberUseCommas (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display commas in numbers with more than three digits to the left of the decimal point.

- 0 No!
- 1 Yes!

|                 |                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Cell, Custom, select Use Commas.                                                 |
| <b>Example</b>  | To display commas in numbers, the command is:<br><code>TableCellNumberUseCommas (State: Yes!)</code> |
| <b>See Also</b> | • <code>TableCellNumberFormat</code>                                                                 |

---

## TableCellNumberUseCurrency

|                   |                                                                                                                                                                                                                           |   |     |   |      |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----|---|------|
| <b>Purpose</b>    | <code>TableCellNumberUseCurrency</code> displays a currency symbol with numbers.                                                                                                                                          |   |     |   |      |
| <b>Syntax</b>     | <code>TableCellNumberUseCurrency</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                     |   |     |   |      |
| <b>Parameters</b> |                                                                                                                                                                                                                           |   |     |   |      |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to display numbers with a currency symbol.<br><table style="margin-left: 40px;"> <tr> <td>0</td> <td>No!</td> </tr> <tr> <td>1</td> <td>Yes!</td> </tr> </table> | 0 | No! | 1 | Yes! |
| 0                 | No!                                                                                                                                                                                                                       |   |     |   |      |
| 1                 | Yes!                                                                                                                                                                                                                      |   |     |   |      |
| <b>Route</b>      | Table, Number Type, Cell, Custom, select Use Currency Symbol.                                                                                                                                                             |   |     |   |      |
| <b>Example</b>    | To display a currency symbol with numbers, the command is:<br><code>TableCellNumberUseCurrency (State: Yes!)</code>                                                                                                       |   |     |   |      |
| <b>See Also</b>   | • <code>TableCellNumberFormat</code>                                                                                                                                                                                      |   |     |   |      |

---

## TableCellUseColumnAttribute

|                   |                                                                                                                                                                                                                                              |   |     |   |      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----|---|------|
| <b>Purpose</b>    | <code>TableCellUseColumnAttribute</code> specifies whether to use the specified column attribute in one or more selected cells.                                                                                                              |   |     |   |      |
| <b>Syntax</b>     | <code>TableCellUseColumnAttribute</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                       |   |     |   |      |
| <b>Parameters</b> |                                                                                                                                                                                                                                              |   |     |   |      |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to use the specified column attribute in the current cell(s).<br><table style="margin-left: 40px;"> <tr> <td>0</td> <td>No!</td> </tr> <tr> <td>1</td> <td>Yes!</td> </tr> </table> | 0 | No! | 1 | Yes! |
| 0                 | No!                                                                                                                                                                                                                                          |   |     |   |      |
| 1                 | Yes!                                                                                                                                                                                                                                         |   |     |   |      |

|                 |                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Format, Cell, Use Column Appearance and Text Size.                                                                            |
| <b>Example</b>  | To use the specified column attribute in the current cell, the command is:<br><code>TableCellUseColumnAttribute (State: Yes!)</code> |
| <b>See Also</b> | ♦ <code>TableCellUseColumnJust</code>                                                                                                |

---

## TableCellUseColumnJust

|                   |                                                                                                                                                                                                                                                |   |     |   |      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----|---|------|
| <b>Purpose</b>    | <code>TableCellUseColumnJust</code> specifies whether to use the current column justification in the current cell.                                                                                                                             |   |     |   |      |
| <b>Syntax</b>     | <code>TableCellUseColumnJust</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                              |   |     |   |      |
| <b>Parameters</b> |                                                                                                                                                                                                                                                |   |     |   |      |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to use the current column justification in the current cell(s).<br><table style="margin-left: 40px;"> <tr> <td>0</td> <td>No!</td> </tr> <tr> <td>1</td> <td>Yes!</td> </tr> </table> | 0 | No! | 1 | Yes! |
| 0                 | No!                                                                                                                                                                                                                                            |   |     |   |      |
| 1                 | Yes!                                                                                                                                                                                                                                           |   |     |   |      |
| <b>Route</b>      | Table, Format, Cell, Use Column Justification.                                                                                                                                                                                                 |   |     |   |      |
| <b>Example</b>    | To use the current column justification in the current cell, the command is:<br><code>TableCellUseColumnJust (State: Yes!)</code>                                                                                                              |   |     |   |      |
| <b>See Also</b>   | ♦ <code>TableCellUseColumnAttribute</code> ♦ <code>TableCellUseColumnNumberType</code>                                                                                                                                                         |   |     |   |      |

---

## TableCellUseColumnNumberType

|                |                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>TableCellUseColumnNumberType</code> specifies whether to use the current column number type in one or more selected cells. |
| <b>Syntax</b>  | <code>TableCellUseColumnNumberType</code> (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                          |

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to use the current column number type in the current cell(s).

- 0 No!
- 1 Yes!

---

**Route** Table, Number Type, Use Column Type.

---

**Example** To use the current column number type in the current cell, the command is:  
TableCellUseColumnNumberType (State: Yes!)

---

**See Also** ♦ TableCellUseColumnAttribute ♦ TableCellUseColumnJust

---

## TableCellUseLineColor

---

**Purpose** TableCellUseLineColor specifies a default line color for lines around one or more selected cells.

---

**Syntax** TableCellUseLineColor ()

---

**Route** Table, Lines/Fill, Current Cell or Selection, deselect Custom Color.

---

**See Also** ♦ LineStyleEdit ♦ TableCellLineColor

---

## TableCellVerticalAlignment

---

**Purpose** TableCellVerticalAlignment specifies vertical alignment for one or more selected cells.

---

**Syntax** TableCellVerticalAlignment (Alignment: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Alignment* Numeric Equivalent or Enumerated Type: Specifies vertical alignment for a cell or blocked cells.

- 1 Top!
- 2 Center!
- 4 Bottom!

---

|                 |                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Format, Cell, select a Vertical Alignment.                                                            |
| <b>Example</b>  | To align text at the bottom, the command is:<br><code>TableCellVerticalAlignment (Alignment: Bottom!)</code> |
| <b>See Also</b> | ♦ <code>TableCellJustification</code>                                                                        |

---

---

## TableColumnAttributeOff

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | <code>TableColumnAttributeOff</code> turns off one or more character attributes in a table column.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Syntax</b>     | <code>TableColumnAttributeOff</code> ({Attributes: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> })                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>Attributes</i> | Numeric Equivalent or Enumerated Type: Specifies an attribute to turn off. Enclose multiple attributes in braces ({}), separated by semicolons.<br><ul style="list-style-type: none"><li>0 ExtraLarge!</li><li>1 VeryLarge!</li><li>2 Large!</li><li>3 Small!</li><li>4 Fine!</li><li>5 Superscript!</li><li>6 Subscript!</li><li>7 Outline!</li><li>8 Italics!</li><li>9 Shadow!</li><li>10 Redline!</li><li>11 DoubleUnderline!</li><li>12 Bold!</li><li>13 Strikeout!</li><li>14 Underline!</li><li>15 SmallCaps!</li><li>16 Every!</li></ul> |

---

|                |                                                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>   | Table, Format, Column, deselect attributes.                                                                                                       |
| <b>Example</b> | To turn off Bold and Italics in a table column, the command is:<br><code>TableColumnAttributeOff</code> ({Attribute: Bold!; Attribute: Italics!}) |

---

---

**See Also**

- `TableColumnAttributeOn`

---

## **TableColumnAttributeOn**

---

**Purpose**

`TableColumnAttributeOn` turns on one or more character attributes in a table column.

**Syntax**

`TableColumnAttributeOn` ({Attribute: *Numeric Equivalent* or *Enumerated Type*})

**Parameters***Attribute*

Numeric Equivalent or Enumerated Type: Specifies an attribute to turn on. Enclose multiple attributes in braces ({}), separated by semicolons.

- 0 ExtraLarge!
- 1 VeryLarge!
- 2 Large!
- 3 Small!
- 4 Fine!
- 5 Superscript!
- 6 Subscript!
- 7 Outline!
- 8 Italics!
- 9 Shadow!
- 10 Redline!
- 11 DoubleUnderline!
- 12 Bold!
- 13 Strikeout!
- 14 Underline!
- 15 SmallCaps!

---

**Route**

Table, Format, Column, select attributes.

---

**Example**

To turn on Bold and Italics in a table column, the command is:

```
TableColumnAttributeOn ({Attribute: Bold!; Attribute: Italics!})
```

---

**See Also**

- `TableColumnAttributeOff`

---

## TableColumnDecimalAlignDigits

---

|                   |                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableColumnDecimalAlignDigits specifies the number of digits between the decimal point and the right edge of cells in the current column.<br>IMPORTANT: This command applies only to cells with decimal align justification. |
| <b>Syntax</b>     | TableColumnDecimalAlignDigits (Digits: <i>Numeric Expression</i> )                                                                                                                                                           |
| <b>Parameters</b> |                                                                                                                                                                                                                              |
| <i>Digits</i>     | Numeric Expression: Specifies the number of digits between the decimal point and the right edge of the cells in the current column.                                                                                          |
| <b>Route</b>      | Table, Format, Column, Justification, Decimal Align, Digits after Decimal, specify a number of digits.                                                                                                                       |
| <b>Example</b>    | To specify three digit places between the decimal point and the right edge of the cells in the current column, the command is:<br><br>TableColumnDecimalAlignDigits (Digits: 3)                                              |
| <b>See Also</b>   | ♦ TableCellJustification ♦ TableColumnDecimalAlignDist<br>♦ TableColumnJustification ♦ TableFormatJustification                                                                                                              |

---

## TableColumnDecimalAlignDist

---

|                   |                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableColumnDecimalAlignDist specifies the distance between the decimal point and the right edge of cells in the current column.<br>IMPORTANT: This command applies only to cells with decimal align justification. |
| <b>Syntax</b>     | TableColumnDecimalAlignDist (Distance: <i>Measurement Expression</i> )                                                                                                                                             |
| <b>Parameters</b> |                                                                                                                                                                                                                    |
| <i>Distance</i>   | Measurement Expression: Specifies the distance between the decimal point and the right edge of cells in the current column.                                                                                        |
| <b>Route</b>      | Table, Format, Column, Justification, Decimal Align, Position from Right, specify a distance.                                                                                                                      |
| <b>Example</b>    | To specify 0.2 inches between the decimal point and the right edge of the cells in the current column, the command is:<br><br>TableColumnDecimalAlignDist (Distance: 0.2")                                         |



---

**See Also**

- ♦ `TableCellJustification` ♦ `TableColumnDecimalAlignDigits`
- ♦ `TableColumnJustification` ♦ `TableFormatJustification`

---

## TableColumnFixedWidth

---

**Purpose**

`TableColumnFixedWidth` designates a column width as fixed.

---

**Syntax**

`TableColumnFixedWidth` (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type: Specifies whether a column has a fixed width.

- 0 No!
- 1 Yes!

---

**Route**

Table, Format, Column, Fixed Width from Column Width group box.

---

**Example**

To designate a column width as fixed, the command is:

```
TableColumnFixedWidth (State: Yes!)
```

---

**See Also**

- ♦ `TableColumnsFixedWidth` ♦ `TableColumnWidth`

---

## TableColumnJustification

---

**Purpose**

`TableColumnJustification` specifies the justification for a column.

---

**Syntax**

`TableColumnJustification` (Mode: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***Mode*

Numeric Equivalent or Enumerated Type: Specifies column justification.

- 0 Left!
- 1 Full!
- 2 Center!
- 3 Right!
- 4 FullAll!
- 5 DecAlign!

---

|                 |                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Format, Column, choose an option from the Justification pop-up list.                                          |
| <b>Example</b>  | To specify right justification for a column, the command is:<br><code>TableColumnJustification (Mode: Right!)</code> |
| <b>See Also</b> | ♦ <code>TableCellJustification</code> ♦ <code>TableFormatJustification</code>                                        |

---

## TableColumnMarginLeft

---

|                   |                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | <code>TableColumnMarginLeft</code> specifies the distance between the left edge of a column and the text in the column. |
| <b>Syntax</b>     | <code>TableColumnMarginLeft</code> (Margin: <i>Measurement Expression</i> )                                             |
| <b>Parameters</b> |                                                                                                                         |
| <i>Margin</i>     | Measurement Expression: Specifies the distance between the left edge of a table column and the text in the column.      |
| <b>Route</b>      | Table, Format, Column, specify a left margin.                                                                           |
| <b>Example</b>    | To specify a left column margin of 0.2 inches, the command is:<br><code>TableColumnMarginLeft (Margin: 0.2")</code>     |
| <b>See Also</b>   | ♦ <code>TableColumnMarginRight</code> ♦ <code>TableMarginLeft</code> ♦ <code>TableMarginRight</code>                    |

---

## TableColumnMarginRight

---

|                   |                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | <code>TableColumnMarginRight</code> specifies the distance between the right edge of a column and the text in the column. |
| <b>Syntax</b>     | <code>TableColumnMarginRight</code> (Margin: <i>Measurement Expression</i> )                                              |
| <b>Parameters</b> |                                                                                                                           |
| <i>Margin</i>     | Measurement Expression: Specifies the distance between the right edge of a table column and the text in the column.       |

|                 |                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Format, Column, specify a right margin.                                                                                              |
| <b>Example</b>  | To specify a right column margin of 0.5 inches, the command is:<br><p style="text-align: center;">TableColumnMarginRight (Margin: 0.5")</p> |
| <b>See Also</b> | ♦ TableColumnMarginLeft ♦ TableMarginLeft ♦ TableMarginRight                                                                                |

---

## TableColumnNumberAlignCurrency

|                   |                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableColumnNumberAlignCurrency aligns currency symbols in a column.                                                                                              |
| <b>Syntax</b>     | TableColumnNumberAlignCurrency (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                     |
| <b>Parameters</b> |                                                                                                                                                                  |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to align currency symbols in the current column.<br><p style="text-align: center;">0 No!<br/>1 Yes!</p> |

|                 |                                                                                                                                                     |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Column, Currency, Custom, Align Currency Symbol.                                                                                |
| <b>Example</b>  | To align currency symbols in the current column, the command is:<br><p style="text-align: center;">TableColumnNumberAlignCurrency (State: Yes!)</p> |
| <b>See Also</b> | ♦ TableColumnNumberSelCurrency                                                                                                                      |

---

## TableColumnNumberDateFormat

|                |                                                                                             |
|----------------|---------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableColumnNumberDateFormat specifies a date format to display dates in the current column. |
| <b>Syntax</b>  | TableColumnNumberDateFormat (DateFmtNum: <i>Numeric Expression</i> )                        |

---

## Parameters

|                   |                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>DateFmtNum</i> | Numeric Expression: Specifies a date format.<br><ol style="list-style-type: none"><li>1 July 5, 1959</li><li>2 7/5/59</li><li>3 Jul 5, 1959</li><li>4 5 July 1959</li><li>5 Sunday, July 5, 1959</li><li>6 12:00 pm</li><li>7 5Jul59</li><li>8 July 5, 1959 (12:00pm)</li><li>9 1959-05-08</li><li>10 7/5</li><li>11 July 5</li><li>12 15:02:02</li></ol> |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|              |                                                                         |
|--------------|-------------------------------------------------------------------------|
| <b>Route</b> | Table, Number Type, Column, Date/Time, Custom, select a Date/Time type. |
|--------------|-------------------------------------------------------------------------|

---

|                |                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To display the date in the format Sunday, July 5, 1959, the command is:<br><code>TableColumnNumberDateFormat (DateFmtNum: 5)</code> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                                        |
|-----------------|----------------------------------------|
| <b>See Also</b> | • <code>TableColumnNumberFormat</code> |
|-----------------|----------------------------------------|

---

## TableColumnNumberDecimalDigits

---

|                |                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>TableColumnNumberDecimalDigits</code> specifies how many digits appear after the decimal point.<br><br>IMPORTANT: This command must be preceded by <code>TableColumnNumberFormat</code> . It is used with the <code>Fixed2!</code> option of <code>TableColumnNumberFormat</code> . |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|               |                                                                                    |
|---------------|------------------------------------------------------------------------------------|
| <b>Syntax</b> | <code>TableColumnNumberDecimalDigits (NumDigits: <i>Numeric Expression</i>)</code> |
|---------------|------------------------------------------------------------------------------------|

---

## Parameters

|                  |                                                                               |
|------------------|-------------------------------------------------------------------------------|
| <i>NumDigits</i> | Numeric Expression: Specifies how many digits appear after the decimal point. |
|------------------|-------------------------------------------------------------------------------|

---

|              |                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------|
| <b>Route</b> | Table, Number Type, Column, Fixed, Custom, specify a number of digits in the Digits after Decimal entry box. |
|--------------|--------------------------------------------------------------------------------------------------------------|

---

|                |                                                                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify three digits to appear after the decimal point, the command sequence is:<br><code>TableColumnNumberFormat (FormatType: Fixed2!)</code><br><code>TableColumnNumberDecimalDigits (NumDigits: 3)</code> |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**See Also** ♦ TableColumnNumberFormat

---

## TableColumnNumberFormat

---

**Purpose** TableColumnNumberFormat specifies a number format for the current column.

**Syntax** TableColumnNumberFormat (FormatType: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*FormatType* Numeric Equivalent or Enumerated Type: Specifies a number format.

- 0 General!
- 1 Integer!
- 2 Fixed2!
- 3 Percent!
- 4 Currency!
- 5 Accounting!
- 6 Commas!
- 7 Scientific!
- 8 DateFormat!
- 9 TextOnly!

**Route** Table, Number Type, Column, select an available type.

---

**Example** To specify currency as the number type, the command is:

TableColumnNumberFormat (FormatType: Currency!)

---

**See Also** ♦ TableColumnNumberAlignCurrency ♦ TableColumnNumberDateFormat  
♦ TableColumnNumberDecimalDigits ♦ TableColumnNumberRound  
♦ TableColumnNumberSelCurrency

---

## TableColumnNumberNegNumber

---

**Purpose** TableColumnNumberNegNumber specifies how to display negative numbers in the current column.

**Syntax** TableColumnNumberNegNumber (NegDisplayType: *Numeric Equivalent* or *Enumerated Type*)

---

---

**Parameters**

*NegDisplayType* Numeric Equivalent or Enumerated Type: Specifies a negative numbers display style.

- 0 Minus!
- 1 Parentheses!
- 2 CRDR!

---

**Route**

Table, Number Type, Column, Custom, select an option from the Negative Numbers group box.

---

**Example**

To display numbers in the Credit/Debit style, the command is:

TableColumnNumberNegNumber (NegDisplayType: CRDR!)

---

## TableColumnNumberRound

---

**Purpose**

TableColumnNumberRound specifies how to convert numbers entered as decimals to integers in the current column.

IMPORTANT: Use this command with the Integer! option of TableColumnNumberFormat.

---

**Syntax**

TableColumnNumberRound (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type: Specifies whether to round off numbers or truncate them when converting to integers.

- 0 No!
- 1 Yes!

---

**Route**

Table, Number Type, Column, Integer, Custom, Round for Calculation.

---

**Example**

To round off numbers after the decimal point when converting them to integers, the command sequence is:

TableColumnNumberFormat (FormatType: Integer!)  
TableColumnNumberRound (State: Yes!)

---

**See Also**

- ♦ TableColumnNumberFormat

---

# TableColumnNumberSelCurrency

---

**Purpose** TableColumnNumberSelCurrency specifies a currency type for the current column.

---

**Syntax** TableColumnNumberSelCurrency (IntlCurrency: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*IntlCurrency* Numeric Equivalent or Enumerated Type: Specifies a currency type.

- 0 DefaultCurrency!
- 1 Dollar!
- 2 Peso!
- 3 FrenchCanada!
- 4 Canada!
- 5 Brazil!
- 6 Denmark!
- 7 Peseta!
- 8 Croatia!
- 9 CzechRepublic!
- 10 SlovakRepublic!
- 11 Finland!
- 12 FinlandIntl!
- 13 France!
- 14 Germany!
- 15 Greece!
- 16 Hungary!
- 17 Iceland!
- 18 Italy!
- 19 Japan!
- 20 Netherlands!
- 21 Norway!
- 22 NorwayIntl!
- 23 Ruble!
- 24 RussianRuble!
- 25 SouthAfrica!
- 26 Sweden!
- 27 Turkey!
- 28 UnitedKingdom!
- 29 Switzerland!
- 30 Austria!
- 31 Belgium!
- 32 Ecuador!
- 33 ElSalvador!
- 34 Guatemala!
- 35 Honduras!
- 36 Paraguay!

- 37 Venezuela!
- 38 Poland!
- 39 Slovenia!

---

**Route** Table, Number Type, Column, Currency, Custom, select a currency type from the Symbol pop-up list.

---

**Example** To specify pesos as the currency type, the command is:

TableColumnNumberSelCurrency (IntCurrency: Peso!)

---

**See Also** ♦ TableColumnNumberAlignCurrency ♦ TableColumnNumberFormat

---

## TableColumnNumberUseCommas

---

**Purpose** TableColumnNumberUseCommas displays commas in numbers with more than three digits to the left of the decimal point.

---

**Syntax** TableColumnNumberUseCommas (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display commas in numbers with more than three digits to the left of the decimal point.

- 0 No!
  - 1 Yes!
- 

**Route** Table, Number Type, Column, select an available type, Custom, Use Commas.

---

**Example** To display commas in numbers, the command is:

TableColumnNumberUseCommas (State: Yes!)

---

**See Also** ♦ TableColumnNumberFormat

---

## TableColumnNumberUseCurrency

---

**Purpose** TableColumnNumberUseCurrency displays a currency symbol with numbers.

---

**Syntax** TableColumnNumberUseCurrency (State: *Numeric Equivalent* or *Enumerated Type*)

---



---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display numbers with a currency symbol.

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

---

**Route** Table, Number Type, Column, select an available type, Custom, Use Currency Symbol.

---

**Example** To display a currency symbol with numbers, the command is:

```
TableColumnNumberUseCurrency (State: Yes!)
```

---

**See Also** ♦ [TableColumnNumberFormat](#)

---

## TableColumnFixedWidth

---

**Purpose** `TableColumnFixedWidth` designates the width of columns in a table as fixed.

---

**Syntax** `TableColumnFixedWidth` (State: *Numeric Equivalent or Enumerated Type*)

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether the width of columns in a table is fixed.

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

---

**Route** Table, Format, Column, Fixed Width.

---

**Example** To designate the width of columns in a table as fixed, the command is:

```
TableColumnFixedWidth (State: Yes!)
```

---

**See Also** ♦ [TableColumnFixedWidth](#)

---

## TableColumnWidth

---

**Purpose** `TableColumnWidth` specifies the width of the current column.

---

**Syntax** `TableColumnWidth (ColumnWidth: Measurement Expression)`

---

**Parameters**

*ColumnWidth* Measurement Expression: Specifies a column width.

---

**Route** Table, Format, Column, specify a column width.

---

**Example** To specify a column width of 1.5 inches, the command is:

`TableColumnWidth (ColumnWidth: 1.5")`

---

**See Also** ♦ `TableColumnFixedWidth` ♦ `TableFormatWidths`

---

---

## TableColumnWidthGrow

**Purpose** `TableColumnWidthGrow` widens a table column.

---

**Syntax** `TableColumnWidthGrow ()`

---

**Route** Ctrl + . (period)

---

---

## TableColumnWidthShrink

**Purpose** `TableColumnWidthShrink` narrows a table column.

---

**Syntax** `TableColumnWidthShrink ()`

---

**Route** Ctrl + , (comma)

---

---

## TableConvertDlg

**Purpose** `TableConvertDlg` displays the Convert Table dialog box.

IMPORTANT: If no text is selected, this command ends the macro or redirects macro execution to the LABEL specified by ONERROR.

|               |                                                                                                                                  |
|---------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | TableConvertDlg ()                                                                                                               |
| <b>Route</b>  | After selecting text contained in either tabular or parallel columns, Table, Create, either Tabular Columns or Parallel Columns. |

---

## TableConvertFrom

|                   |                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableConvertFrom converts selected text in tabular or parallel columns to a table.                                                                                                                                           |
| <b>Syntax</b>     | TableConvertFrom (Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                |
| <b>Parameters</b> |                                                                                                                                                                                                                              |
| <i>Type</i>       | Numeric Equivalent or Enumerated Type: Specifies whether text to convert is in tabular columns (the text in each column is separated by tabs or indents) or parallel columns.<br><br>0 TabularColumns!<br>1 ParallelColumns! |
| <b>Route</b>      | Table, Create, select an option.                                                                                                                                                                                             |
| <b>Example</b>    | To convert text from tabular columns to a table, the command is:<br><br>TableConvertFrom (Type: TabularColumns!)                                                                                                             |
| <b>See Also</b>   | ♦ TableCreate                                                                                                                                                                                                                |

---

## TableCopy

|                |                                                                                                                                                                                                                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableCopy copies the contents of selected cells, the current column, or the current row.<br><br><b>IMPORTANT:</b> This command must be followed by TableMoveModeCancel or TableMoveModeEnd. Selected cells can be inserted into the table using TableMoveModeEnd.<br><br><b>IMPORTANT:</b> To record this command, you must first select two or more cells. |
| <b>Syntax</b>  | TableCopy (Copy: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                      |

---

## Parameters

*Copy* Numeric Equivalent or Enumerated Type: Specifies whether to copy selected cells, the current row, or the current column.

- 1 Block!
- 2 Row!
- 3 Column!

---

**Route** Select two or more cells, Edit, Copy, select an option.

---

**Example** To copy the contents of the current column, the command is:

TableCopy (Copy: Column!)

---

**See Also** ♦ TableMove

---

## TableCopyFormula

---

**Purpose** TableCopyFormula copies a formula from the current cell to other cells.

---

**Syntax** TableCopyFormula (CellIDCntToCell: *Numeric Equivalent* or *Enumerated Type*, Where: *Character Expression* or *Numeric Expression*)

---

## Parameters

*CellIDCntToCell* Numeric Equivalent or Enumerated Type: Specifies where to move the insertion point.

- 0 Down!
- 3 Right!
- 4 Cell!

*Where* Character Expression or Numeric Expression: Specifies the cell address if Cell! is used, or the number of times to copy the formula if Down! or Right! are used.

---

**Route** Table, Copy Formula.

---

**Example** To copy the formula from the current cell down one cell, the command is:

TableCopyFormula (CellIDCntToCell: Down!; Where: 1)

---

## TableCreate

|                   |                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableCreate creates a table.                                                                             |
| <b>Syntax</b>     | TableCreate (Columns: <i>Numeric Expression</i> ; Rows: <i>Numeric Expression</i> )                      |
| <b>Parameters</b> |                                                                                                          |
| <i>Columns</i>    | Numeric Expression: Specifies the number of columns to create.                                           |
| <i>Rows</i>       | Numeric Expression: Specifies the number of rows to create.                                              |
| <b>Route</b>      | Table, Create, specify number of columns and rows.                                                       |
| <b>Example</b>    | To create a table with three columns and five rows, the command is:<br>TableCreate (Columns: 3; Rows: 5) |
| <b>See Also</b>   | ♦ TableConvertFrom ♦ TableJoin ♦ TableSplit                                                              |

---

## TableCreateDlg

|                 |                                                   |
|-----------------|---------------------------------------------------|
| <b>Purpose</b>  | TableCreateDlg opens the Create Table dialog box. |
| <b>Syntax</b>   | TableCreateDlg ()                                 |
| <b>Route</b>    | Table, Create.                                    |
| <b>See Also</b> | ♦ TableCreate                                     |

---

## TableCreateNewDlg

|                 |                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | TableCreateNewDlg displays the Create Table dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | TableCreateNewDlg ()                                                                                                                                            |
| <b>See Also</b> | ♦ TableCreate                                                                                                                                                   |

---

## TableDataFill

---

|                |                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableDataFill fills a cell with a formula or data.<br><br>IMPORTANT: You must first select at least two cells. The first selected cell must contain data or a formula, and at least one selected cell must be empty. |
| <b>Syntax</b>  | TableDataFill ()                                                                                                                                                                                                     |
| <b>Route</b>   | Table, Data Fill.                                                                                                                                                                                                    |

---

## TableDecimalAlignmentDigits

---

|                   |                                                                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableDecimalAlignmentDigits specifies the number of digits between the decimal point and the right edge of cells in the current table.<br><br>IMPORTANT: This command applies only to cells with decimal align justification. |
| <b>Syntax</b>     | TableDecimalAlignmentDigits (Number: <i>Numeric Expression</i> )                                                                                                                                                              |
| <b>Parameters</b> |                                                                                                                                                                                                                               |
| <i>Number</i>     | Numeric Expression: Specifies the number of digits between the decimal point and the right edge of cells.                                                                                                                     |
| <b>Route</b>      | Table, Format, Table, Justification pop-up list, Decimal Align, specify a number of digits after the decimal.                                                                                                                 |
| <b>Example</b>    | To specify three digit places between the decimal point and the right edge of cells in the current table, the command is:<br><br>TableDecimalAlignmentDigits (Number: 3)                                                      |
| <b>See Also</b>   | ♦ TableDecimalAlignmentDistance ♦ TableFormatJustification                                                                                                                                                                    |

---

## TableDecimalAlignmentDistance

---

|                |                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableDecimalAlignmentDistance specifies the distance between the decimal point and the right edge of cells in the current table.<br><br>IMPORTANT: This command applies only to cells with decimal align justification. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                   |                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | TableDecimalAlignmentDistance (Position: <i>Measurement Expression</i> )                                                                                                |
| <b>Parameters</b> |                                                                                                                                                                         |
| <i>Position</i>   | Measurement Expression: Specifies the distance between the decimal point and the right edge of cells in the current table.                                              |
| <b>Route</b>      | Table, Format, Table, Justification, Decimal Align, Position from Right, specify a distance.                                                                            |
| <b>Example</b>    | To specify 0.2 inches between the decimal point and the right edge of cells in the current table, the command is:<br><br>TableDecimalAlignmentDistance (Position: 0.2") |
| <b>See Also</b>   | ♦ TableDecimalAlignmentDigits ♦ TableFormatJustification                                                                                                                |

## TableDefaultLine

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableDefaultLine changes the default line style for a table.                                                                                                                                                                                                                                                                                                                                                         |
| <b>Syntax</b>     | TableDefaultLine (Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                                                                       |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>Style</i>      | Numeric Equivalent or Enumerated Type: Specifies the line style. This parameter also accepts a user defined graphic line style, enclosed in double quotation marks.<br><br>0 SingleLine!<br>1 DoubleLine!<br>2 DashedLine!<br>3 DottedLine!<br>4 ThickLine!<br>5 ExtraThickLine!<br>6 ThinThickLine!<br>7 ThickThinLine!<br>8 ButtonTopLeftLine!<br>9 ButtonBottomRightLine!<br>126 TableDefaultLine!<br>127 NoLine! |
| <b>Route</b>      | Table, Lines/Fill, Table, select a style from the Default Line Style palette or pop-up list.                                                                                                                                                                                                                                                                                                                         |

---

**Example** To specify a dotted line as the default table line style, the command is:

TableDefaultLine (Style: DottedLine!)

---

**See Also** ♦ TableCellLine

---

## TableDefaultLineColor

---

**Purpose** TableDefaultLineColor specifies the default line color for a table.

---

**Syntax** TableDefaultLineColor (ColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*; Shade: *Numeric Expression*)

---

### Parameters

*ColorName* Character Expression: Specifies a predefined color name.

*Red* Numeric Expression: Specifies the amount of red in a custom color. Values range from 0 to 255.

*Green* Numeric Expression: Specifies the amount of green in a custom color. Values range from 0 to 255.

*Blue* Numeric Expression: Specifies the amount of blue in a custom color. Values range from 0 to 255.

*Shade* Numeric Expression: Specifies the shading percentage in a custom color. Values range from 1 to 100.

---

**Route** Table, Lines/Fill, Table, Custom Color, choose a color from the pop-up palette.

---

**Example** To specify magenta default table lines, the command is:

TableDefaultLineColor (ColorName: "Magenta"; Red: 255; Green: 0; Blue: 255; Shade: 100)

---

**See Also** ♦ TableCellLineColor

---

## TableDeleteBlock

---

**Purpose** TableDeleteBlock deletes the contents of selected cells.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.



---

|                 |                                                      |
|-----------------|------------------------------------------------------|
| <b>Syntax</b>   | TableDeleteBlock ()                                  |
| <b>See Also</b> | ♦ TableDeleteColumn ♦ TableDeleteRow ♦ TableUndelete |

---

## TableDeleteColumn

---

|                   |                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableDeleteColumn deletes one or more table columns.                                                                                                                                                       |
| <b>Syntax</b>     | TableDeleteColumn (Number: <i>Numeric Expression</i> )                                                                                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                                            |
| <i>Number</i>     | Numeric Expression: Specifies the number of columns to delete, beginning with the current column and moving left. If you omit this parameter, this command deletes the current column or selected columns. |
| <b>Route</b>      | Table, Delete, Columns, specify a number of columns to delete.                                                                                                                                             |
| <b>Example</b>    | To delete one column, the command is:<br><br>TableDeleteColumn (Number: 1)                                                                                                                                 |
| <b>See Also</b>   | ♦ TableDeleteBlock ♦ TableDeleteRow ♦ TableUndelete                                                                                                                                                        |

---

## TableDeleteDlg

---

|                 |                                                |
|-----------------|------------------------------------------------|
| <b>Purpose</b>  | TableDeleteDlg displays the Delete dialog box. |
| <b>Syntax</b>   | TableDeleteDlg ()                              |
| <b>Route</b>    | Table, Delete.                                 |
| <b>See Also</b> | ♦ TableDeleteColumn ♦ TableDeleteRow           |

---

---

## TableDeleteRow

---

**Purpose** TableDeleteRow deletes one or more table rows.

**Syntax** TableDeleteRow (Number: *Numeric Expression*)

### Parameters

*Number* Numeric Expression: Specifies the number of rows to delete, beginning with the current row and moving down.

**Route** Table, Delete, Rows, specify the number of rows to delete.

**Example** To delete five table rows, the command is:

TableDeleteRow (Number: 5)

**See Also** ♦ TableDeleteColumn ♦ TableDeleteDlg

---

---

## TableDeleteTable

---

**Purpose** TableDeleteTable deletes a table or its contents, or it converts a table's cells to tabs or merge codes.

**Syntax** TableDeleteTable (DelWhat: *Numeric Equivalent or Enumerated Type*)

### Parameters

*DelWhat* Numeric Equivalent or Enumerated Type: Performs according to parameter options: EntireTable! deletes a table; ConvertOnly! deletes only cell contents; ConvertToTabs! converts cells to tabs; ConvertToMerge! converts the cells of each row to ENDFIELD codes; ConvertToMergeNames! converts the cells in the first row to FIELDNAMES, and the cells in the remaining rows to ENDFIELD codes.

- 0 EntireTable!
- 1 ConvertOnly!
- 2 ConvertToTabs!
- 3 ConvertToMerge!
- 4 ConvertToMergeNames!

**Route** With a table selected, Table, Delete.

**Example** To convert cells to tabs, the command is:

TableDeleteTable (DelWhat: ConvertToTabs!)

---

---

## TableDisableCellLocks

---

|                   |                                                                                     |
|-------------------|-------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableDisableCellLocks specifies whether to turn off cell locks.                     |
| <b>Syntax</b>     | TableDisableCellLocks (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                     |
| <i>State</i>      | Numeric Equivalent or Enumerated Type<br>0 No!<br>1 Yes!                            |
| <b>Route</b>      | Table, Format, Table, Disable Cell Locks.                                           |
| <b>Example</b>    | To disable cell locks, the command is:<br>TableDisableCellLocks (State: Yes!)       |

---

## TableEdit

---

|                 |                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | TableEdit initiates editing changes to a table.<br><br>IMPORTANT: If the insertion point is not in a table, this command searches backward for a table to edit. If no table is found, it then searches forward for a table to edit. If no table is found, a NOT FOUND error is returned. |
| <b>Syntax</b>   | TableEdit ()                                                                                                                                                                                                                                                                             |
| <b>See Also</b> | ♦ TableCreate                                                                                                                                                                                                                                                                            |

---

## TableFill

---

|                |                                                                           |
|----------------|---------------------------------------------------------------------------|
| <b>Purpose</b> | TableFill fills a table with the specified fill style.                    |
| <b>Syntax</b>  | TableFill (Pattern: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

---

---

## Parameters

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| <i>Pattern</i> | Numeric Equivalent or Enumerated Type: Specifies a fill style for a table. |
|                | 0 Fill10!                                                                  |
|                | 1 Fill20!                                                                  |
|                | 2 Fill30!                                                                  |
|                | 3 Fill40!                                                                  |
|                | 4 Fill50!                                                                  |
|                | 5 Fill60!                                                                  |
|                | 6 Fill70!                                                                  |
|                | 7 Fill80!                                                                  |
|                | 8 Fill90!                                                                  |
|                | 9 Fill100!                                                                 |
|                | 10 FillButton!                                                             |
|                | 127 NoFill!                                                                |

---

|              |                                                                           |
|--------------|---------------------------------------------------------------------------|
| <b>Route</b> | Table, Lines/Fill, Table, select a style from the Fill Options group box. |
|--------------|---------------------------------------------------------------------------|

---

|                |                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------|
| <b>Example</b> | To specify 20% shading as the fill style for a table, the command is:<br>TableFill (Pattern: Fill20!) |
|----------------|-------------------------------------------------------------------------------------------------------|

---

|                 |                      |
|-----------------|----------------------|
| <b>See Also</b> | ♦ TableCellFillStyle |
|-----------------|----------------------|

---

## TableFormatDlg

---

|                |                                             |
|----------------|---------------------------------------------|
| <b>Purpose</b> | TableFormatDlg opens the Format dialog box. |
|----------------|---------------------------------------------|

---

|               |                   |
|---------------|-------------------|
| <b>Syntax</b> | TableFormatDlg () |
|---------------|-------------------|

---

|              |                |
|--------------|----------------|
| <b>Route</b> | Table, Format. |
|--------------|----------------|

---

## TableFormatJustification

---

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <b>Purpose</b> | TableFormatJustification specifies justification for a table. |
|----------------|---------------------------------------------------------------|

---

|               |                                                                                                |
|---------------|------------------------------------------------------------------------------------------------|
| <b>Syntax</b> | TableFormatJustification (Justification: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|------------------------------------------------------------------------------------------------|

---

**Parameters****Justification**

Numeric Equivalent or Enumerated Type: Specifies justification for a table.

- 0 Left!
- 1 Full!
- 2 Center!
- 3 Right!
- 4 FullAll!
- 5 DecAlign!

---

**Route**

Table, Format, Table, Justification, select a justification option.

---

**Example**

To specify center justification for a table, the command is:

TableFormatJustification (Justification: Center!)

---

**See Also**

♦ TableCellJustification ♦ TableColumnJustification

---

## TableFormatWidths

---

**Purpose**

TableFormatWidths specifies column width in a table.

---

**Syntax**

TableFormatWidths (Width: *Numeric Expression*)

---

**Parameters****Width**

Numeric Expression: Specifies column width.

---

**Route**

Table, Format, Table, specify a column width.

---

**Example**

To set column width to 1 inch, the command is:

TableFormatWidths (Width: 1")

---

## TableFormula

---

**Purpose**

TableFormula specifies a formula to insert in the current cell.

---

**Syntax**

TableFormula (Formula: *Character Expression*)

---

**Parameters**

*Formula* Character Expression: Specifies a table formula.

---

**Route** Table, Formula Bar, Functions, select a function.

---

**Example** To calculate the average of a group of cells named Sales, the command is:

TableFormula (Formula: "AVE(Sales)")

---

**See Also** ♦ TableNameAdd

---

---

## TableFormulaCBar

**Purpose** TableFormulaCBar displays the Table Formula Control Bar.

---

**Syntax** TableFormulaCBar ()

---

**Route** Table, Formula Bar.

---

---

## TabLeft

**Purpose** TabLeft inserts a Hard Left Tab at the insertion point.

---

**Syntax** TabLeft (Type: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Type* Numeric Equivalent or Enumerated Type: Specifies whether to include a dot leader.

- 0 Normal!
  - 1 DotLeader!
- 

**Route** Layout, Line, Other Codes, select Left under Hard Tab Codes or Hard Tab Codes with Dot Leaders.

---

**Example** To insert a left tab with a dot leader, the command is:

TabLeft (Type: DotLeader!)

---

---

**See Also** ♦ TabCenter ♦ TabDecimal ♦ TabRight

---

## TableGridLinesToggle

---

**Purpose** TableGridLinesToggle specifies whether grid lines are on.

---

**Syntax** TableGridLinesToggle ()

---

**Route** View, Table Gridlines.

---

---

## TableGuides

---

**Purpose** TableGuides specifies whether Table Guides are on.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** TableGuides (State: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*State* Numeric Equivalent or Enumerated Type

- 0 Off!
- 1 On!

---

**Example** To turn on Table Guides, the command is:

TableGuides (State: On!)

---

**See Also** ♦ ?TableGuides

---

---

## TableHardColumnBreak

---

**Purpose** TableHardColumnBreak inserts a column break and continues the table in the next column.

|                 |                                   |
|-----------------|-----------------------------------|
| <b>Syntax</b>   | TableHardColumnBreak ()           |
| <b>Route</b>    | Layout, Columns, Column Break.    |
| <b>See Also</b> | ♦ PosTableRowBegin ♦ TableHardRow |

---

## TableHardRow

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>Purpose</b>  | TableHardRow inserts a page break and continues the table on the next page. |
| <b>Syntax</b>   | TableHardRow ()                                                             |
| <b>Route</b>    | Press Ctrl+Enter while the insertion point is in a table.                   |
| <b>See Also</b> | ♦ PosTableRowBegin ♦ TableHardColumnBreak                                   |

---

## TableHeader

|                   |                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableHeader specifies the current row as a header row.                                                              |
| <b>Syntax</b>     | TableHeader (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                           |
| <b>Parameters</b> |                                                                                                                     |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to use the current row as a header row.<br>0 No!<br>1 Yes! |
| <b>Route</b>      | Table, Format, Row, Header Row.                                                                                     |
| <b>Example</b>    | To make the current row a header row, the command is:<br>TableHeader (State: Yes!)                                  |



---

## TableInsertColumn

|                   |                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableInsertColumn inserts a table column before or after the insertion point.                                           |
| <b>Syntax</b>     | TableInsertColumn (Number: <i>Numeric Expression</i> ; InsertPos: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
| <b>Parameters</b> |                                                                                                                         |
| <i>Number</i>     | Numeric Expression: Specifies how many columns to insert.                                                               |
| <i>InsertPos</i>  | Numeric Equivalent or Enumerated Type: Determines where to insert the column(s).<br>0 Before!<br>1 After!               |
| <b>Route</b>      | Table, Insert, Columns, specify the number of columns and where to insert them.                                         |
| <b>Example</b>    | To insert two columns after the current column, the command is:<br><br>TableInsertColumn (Number: 2; InsertPos: After!) |
| <b>See Also</b>   | ♦ TableInsertRow                                                                                                        |

---

## TableInsertDlg

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| <b>Purpose</b>  | TableInsertDlg displays the Insert Columns/Rows dialog box. |
| <b>Syntax</b>   | TableInsertDlg ()                                           |
| <b>Route</b>    | Table, Insert.                                              |
| <b>See Also</b> | ♦ TableInsertColumn ♦ TableInsertRow                        |

---

## TableInsertOneRow

|                |                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableInsertOneRow inserts a new row above the row containing the insertion point.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Syntax** TableInsertOneRow ()

---

## TableInsertRow

---

**Purpose** TableInsertRow inserts a specified number of rows.

---

**Syntax** TableInsertRow (NumberOfRows: *Numeric Expression*; InsertPosition: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*NumberOfRows* Numeric Expression: Specifies the number of rows to insert.

*InsertPosition* Numeric Equivalent or Enumerated Type: Specifies where to insert the row(s).

0 Before!

1 After!

---

**Route** Table, Insert, Rows, specify the number of rows and where to add them.

---

**Example** To insert a specified number of table rows above the insertion point, the command is:

TableInsertRow (NumberOfRows: 3; InsertPosition: Before!)

---

**See Also** ♦ TableAppendRow ♦ TableInsertColumn ♦ TableInsertDlg

---

## TableJoin

---

**Purpose** TableJoin joins the current table and the table immediately following it into one table. IMPORTANT: The tables must have the same number of columns.

---

**Syntax** TableJoin ()

---

**Route** Table, Join, Table.

---

**See Also** ♦ TableSplit

---

## TableJoinCells

|                |                                                             |
|----------------|-------------------------------------------------------------|
| <b>Purpose</b> | TableJoinCells joins selected table cells to form one cell. |
| <b>Syntax</b>  | TableJoinCells ()                                           |
| <b>Route</b>   | Select cells, Table, Join, Cell.                            |

---

## TableLineFillDlg

|                |                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableLineFillDlg displays the Table Lines/Fill dialog box.<br>IMPORTANT: The insertion point must be in a table. |
| <b>Syntax</b>  | TableLineFillDlg ()                                                                                              |
| <b>Route</b>   | Table, Lines/Fill.                                                                                               |

---

## TableMarginLeft

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableMarginLeft specifies the distance from the left edge of each cell to the text in the cell for the current table. |
| <b>Syntax</b>     | TableMarginLeft (Margin: <i>Measurement Expression</i> )                                                              |
| <b>Parameters</b> |                                                                                                                       |
| <i>Margin</i>     | Measurement Expression: Specifies the distance from the left edge of each cell to the text in the cell.               |
| <b>Route</b>      | Table, Format, Table, Column Margins, Left, specify a measurement.                                                    |
| <b>Example</b>    | To specify a left margin of 0.1 inches for the current table, the command is:<br>TableMarginLeft (Margin: 0.1")       |
| <b>See Also</b>   | ♦ TableColumnMarginLeft ♦ TableColumnMarginRight ♦ TableMarginRight                                                   |

---

## TableMarginRight

---

|                   |                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableMarginRight specifies the distance from the right edge of each cell to the text in the cell for the current table.           |
| <b>Syntax</b>     | TableMarginRight (Margin: <i>Measurement Expression</i> )                                                                         |
| <b>Parameters</b> |                                                                                                                                   |
| <i>Margin</i>     | Measurement Expression: Specifies the distance from the right edge of each cell to the text in the cell.                          |
| <b>Route</b>      | Table, Format, Table, Column Margins, Right, specify a measurement.                                                               |
| <b>Example</b>    | To specify a right cell margin of 0.1 inches for the current table, the command is:<br><pre>TableMarginRight (Margin: 0.1")</pre> |
| <b>See Also</b>   | ♦ TableColumnMarginLeft ♦ TableColumnMarginRight ♦ TableMarginLeft                                                                |

---

## TableMove

---

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableMove moves the contents of a cell, blocked cells, current column, or current row.<br><br>IMPORTANT: To record this command, you must first select two or more cells.<br><br>IMPORTANT: After TableMove is executed, the contents of the cell(s) can be inserted into the table with TableMoveModeEnd. TableMove must be used in conjunction with other commands, such as TableMoveModeCancel and TableMoveModeEnd. |
| <b>Syntax</b>     | TableMove (Move: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                                                                                                                                                                                                  |
| <b>Parameters</b> |                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>Move</i>       | Numeric Equivalent or Enumerated Type: Specifies the source of the contents to move.<br><br><ol style="list-style-type: none"><li>1 Block!</li><li>2 Row!</li><li>3 Column!</li><li>4 Cell!</li></ol>                                                                                                                                                                                                                   |

|                 |                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Select two or more cells, Edit, Cut, select an option.                                                                      |
| <b>Example</b>  | To move the contents of the current column, the command is:<br><p style="text-align: center;">TableMove (Move: Column!)</p> |
| <b>See Also</b> | ♦ TableCopy                                                                                                                 |

---

## TableMoveCopyDlg

|                 |                                             |
|-----------------|---------------------------------------------|
| <b>Purpose</b>  | TableMoveCopyDlg opens the Move dialog box. |
| <b>Syntax</b>   | TableMoveCopyDlg ()                         |
| <b>Route</b>    | Layout, Tables, Edit, Move/Copy.            |
| <b>See Also</b> | ♦ TableCopy ♦ TableMove                     |

---

## TableNameAdd

|                |                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableNameAdd names a cell, block of cells, row, column, or table.                                                |
| <b>Syntax</b>  | TableNameAdd (RefType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Name: <i>Character Expression</i> ) |

### Parameters

*RefType* Numeric Equivalent or Enumerated Type: Specifies cells to name.

- 1 Block!
- 2 Row!
- 3 Column!
- 4 Cell!
- 5 Table!
- 6 FloatingCell!
- 7 CellsDown!
- 8 CellsRight!

*Name* Character Expression: Specifies a name.

|                 |                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Names, Create, enter a name and select a range of cells.                                                               |
| <b>Example</b>  | To name a block of cells Sales Figures, the command is:<br><code>TableNameAdd (RefType: Block!; Name: "Sales Figures")</code> |
| <b>See Also</b> | ♦ <code>TableNameDelete</code>                                                                                                |

---

## TableNameChange

|                   |                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | <code>TableNameChange</code> renames an existing table.                                                                                                       |
| <b>Syntax</b>     | <code>TableNameChange (OldName: <i>Character Expression</i>; NewName: <i>Character Expression</i>; NewRef: <i>Character Expression</i>)</code>                |
| <b>Parameters</b> |                                                                                                                                                               |
| <i>OldName</i>    | Character Expression (optional): Specifies a table name to change.                                                                                            |
| <i>NewName</i>    | Character Expression (optional): Specifies a new name for a table.                                                                                            |
| <i>NewRef</i>     | Character Expression (optional): Specifies a new reference.                                                                                                   |
| <b>Route</b>      | Table, Names, Edit, enter a name.                                                                                                                             |
| <b>Example</b>    | To specify a table named “Old Table” and rename it “New Table,” the command is:<br><code>TableNameChange (OldName: "Old Table"; NewName: "New Table");</code> |

---

## TableNameCreateDlg

|                |                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>TableNameCreateDlg</code> displays the Create Name dialog box.<br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | <code>TableNameCreateDlg ()</code>                                                                                                                                              |

---

## TableNameDelete

|                   |                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableNameDelete deletes a table name.                                                                                               |
| <b>Syntax</b>     | TableNameDelete (TableName: <i>Character Expression</i> )                                                                           |
| <b>Parameters</b> |                                                                                                                                     |
| <i>TableName</i>  | Character Expression: Specifies the name of a table to delete. The name can be preceded by the name of a table and a period.        |
| <b>Route</b>      | Table, Names, select a name, Delete.                                                                                                |
| <b>Example</b>    | To delete the table name Sales in a table named Table_C, the command is:<br><pre>TableNameDelete (TableName: "Table_C.Sales")</pre> |
| <b>See Also</b>   | ♦ TableNameAdd                                                                                                                      |

---

## TableNameListDlg

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| <b>Purpose</b>  | TableNameListDlg opens the Table Names in Current Document dialog box. |
| <b>Syntax</b>   | TableNameListDlg ()                                                    |
| <b>Route</b>    | Table, Names.                                                          |
| <b>See Also</b> | ♦ TableNameAdd                                                         |

---

## TableNumberAlignCurrency

|                   |                                                                                                                                                                                            |   |     |   |      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----|---|------|
| <b>Purpose</b>    | TableNumberAlignCurrency aligns currency symbols in a table.                                                                                                                               |   |     |   |      |
| <b>Syntax</b>     | TableNumberAlignCurrency (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                     |   |     |   |      |
| <b>Parameters</b> |                                                                                                                                                                                            |   |     |   |      |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether to align currency symbols in the current table.<br><table><tr><td>0</td><td>No!</td></tr><tr><td>1</td><td>Yes!</td></tr></table> | 0 | No! | 1 | Yes! |
| 0                 | No!                                                                                                                                                                                        |   |     |   |      |
| 1                 | Yes!                                                                                                                                                                                       |   |     |   |      |

|                 |                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Table, Currency, Custom, Align Currency Symbol.                                                    |
| <b>Example</b>  | To align currency symbols in the current table, the command is:<br><code>TableNumberAlignCurrency (State: Yes!)</code> |
| <b>See Also</b> | ♦ <code>TableNumberSelectCurrency</code>                                                                               |

## TableNumberDateFormat

|                |                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>TableNumberDateFormat</code> specifies a date format to display dates in the current table. |
| <b>Syntax</b>  | <code>TableNumberDateFormat (DateFmtNum: <i>Numeric Expression</i>)</code>                        |

### Parameters

|                   |                                                                                                                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>DateFmtNum</i> | Numeric Expression: Specifies a date format.<br><ol style="list-style-type: none"> <li>1 July 5, 1959</li> <li>2 7/5/59</li> <li>3 Jul 5, 1959</li> <li>4 5 July 1959</li> <li>5 Sunday, July 5, 1959</li> <li>6 12:00 pm</li> <li>7 05Jul59</li> <li>8 July 5, 1959 (12:00pm)</li> </ol> |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                 |                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Table, Date/Time, Custom, select a type.                                                                  |
| <b>Example</b>  | To display the date in the format Sunday, July 5, 1959, the command is:<br><code>TableNumberDateFormat (DateFmtNum: 5)</code> |
| <b>See Also</b> | ♦ <code>TableNumberFormat</code>                                                                                              |

## TableNumberDecimalDigits

|                |                                                                                                                                                                                            |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <code>TableNumberDecimalDigits</code> specifies how many digits appear after the decimal point.<br>IMPORTANT: Use with the <code>Fixed2!</code> option of <code>TableNumberFormat</code> . |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



|                   |                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | TableNumberDecimalDigits (NumDigits: <i>Numeric Expression</i> )                                                                                                                    |
| <b>Parameters</b> |                                                                                                                                                                                     |
| <i>NumDigits</i>  | Numeric Expression: Specifies the number of digits that appear after the decimal point.                                                                                             |
| <b>Route</b>      | Table, Number Type, Table, Integer, Custom, specify a number in the Decimal Amounts text box.                                                                                       |
| <b>Example</b>    | To specify that three digits appear after the decimal point, the command sequence is:<br><pre>TableNumberFormat (FormatType: Fixed2!) TableNumberDecimalDigits (NumDigits: 3)</pre> |
| <b>See Also</b>   | ♦ TableNumberFormat                                                                                                                                                                 |

---

## TableNumberFormat

|                   |                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableNumberFormat specifies a number format for the current table.                                                                                                                                                           |
| <b>Syntax</b>     | TableNumberFormat (FormatType: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                         |
| <b>Parameters</b> |                                                                                                                                                                                                                              |
| <i>FormatType</i> | Numeric Equivalent or Enumerated Type: Specifies a number format for the current table.<br><pre>0 General! 1 Integer! 2 Fixed2! 3 Percent! 4 Currency! 5 Accounting! 6 Commas! 7 Scientific! 8 DateFormat! 9 TextOnly!</pre> |
| <b>Route</b>      | Table, Number Type, Table, select a format.                                                                                                                                                                                  |
| <b>Example</b>    | To specify currency as the number type, the command is:<br><pre>TableNumberFormat (FormatType: Currency!)</pre>                                                                                                              |

---

**See Also**

- `TableColumnNumberAlignCurrency`
- `TableColumnNumberDateFormat`
- `TableColumnNumberDecimalDigits`
- `TableColumnNumberRound`
- `TableColumnNumberSelCurrency`

---

## TableNumberNegativeNumber

**Purpose**

`TableNumberNegativeNumber` specifies how to display negative numbers in the current table.

**Syntax**

`TableNumberNegativeNumber` (NegDisplay: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*NegDisplay*

Numeric Equivalent or Enumerated Type: Specifies a display style for negative numbers.

- 0 Minus!
- 1 Parentheses!
- 2 CRDR!

**Route**

Table, Number Type, Table, select a type, Custom, select an option from the Negative Numbers group box.

**Example**

To display numbers using the Credit/Debit method, the command is:

```
TableNumberNegativeNumber (NegDisplay: CRDR!)
```

---

## TableNumberRound

**Purpose**

`TableNumberRound` specifies how to convert decimal numbers to integers. IMPORTANT: Use with the Integer! option of `TableNumberFormat`.

**Syntax**

`TableNumberRound` (State: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*State*

Numeric Equivalent or Enumerated Type: Specifies whether to round off or truncate numbers when converting them to integers.

- 0 No!
- 1 Yes!

|                 |                                                                                                                                                                                     |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Route</b>    | Table, Number Type, Table, Integer, Custom, Round for Calculation.                                                                                                                  |
| <b>Example</b>  | To round off numbers after the decimal point when converting to integers, the command is:<br><br><pre>TableNumberFormat (FormatType: Integer!) TableNumberRound (State: Yes!)</pre> |
| <b>See Also</b> | ♦ TableNumberFormat                                                                                                                                                                 |

---

## TableNumberSelectCurrency

|                |                                                                                                |
|----------------|------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TableNumberSelectCurrency specifies a currency type for the current table.                     |
| <b>Syntax</b>  | TableNumberSelectCurrency (IntlCurrency: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |

### Parameters

*IntlCurrency* Numeric Equivalent or Enumerated Type: Specifies a currency type.

- 0 Default Currency!
- 1 Dollar!
- 2 Peso!
- 3 FrenchCanada!
- 4 Canada!
- 5 Brazil!
- 6 Denmark!
- 7 Peseta!
- 8 Croatia!
- 9 CzechRepublic!
- 10 SlovakRepublic!
- 11 Finland!
- 12 FinlandIntl!
- 13 France!
- 14 Germany!
- 15 Greece!
- 16 Hungary!
- 17 Iceland!
- 18 Italy!
- 19 Japan!
- 20 Netherlands!
- 21 Norway!
- 22 NorwayIntl!
- 23 Ruble!
- 24 RussianRuble!
- 25 SouthAfrica!

- 26 Sweden!
- 27 Turkey!
- 28 UnitedKingdom!
- 29 Switzerland!
- 30 Austria!
- 31 Belgium!
- 32 Ecuador!
- 33 ElSalvador!
- 34 Guatemala!
- 35 Honduras!
- 36 Paraguay!
- 37 Venezuela!
- 38 Poland!
- 39 Slovenia!

---

**Route** Table, Number Type, Table, Currency, Custom, select an option in the Symbol pop-up list.

---

**Example** To specify pesos as the currency type, the command is:

`TableNumberSelectCurrency (IntlCurrency: Peso!)`

---

**See Also** ♦ `TableNumberAlignCurrency`

---

## TableNumberTypeDlg

---

**Purpose** `TableNumberTypeDlg` displays the Number Type dialog box.

---

**Syntax** `TableNumberTypeDlg ()`

---

**Route** Table, Number Type.

---

## TableNumberUseCommas

---

**Purpose** `TableNumberUseCommas` displays commas in numbers with more than three digits to the left of the decimal point.

---

**Syntax** `TableNumberUseCommas (State: Numeric Equivalent or Enumerated Type)`

---

---

## Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display commas in numbers with more than three digits to the left of the decimal point.

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

---

**Route** Table, Number Type, Table, select an available type, Custom, Use Commas.

---

**Example** To display commas in numbers, the command is:

```
TableNumberUseCommas (State: Yes!)
```

---

**See Also** ♦ TableNumberFormat

---

## TableNumberUseCurrency

---

**Purpose** TableNumberUseCurrency displays a currency symbol with numbers.

---

**Syntax** TableNumberUseCurrency (State: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*State* Numeric Equivalent or Enumerated Type: Specifies whether to display numbers with a currency symbol.

|   |      |
|---|------|
| 0 | No!  |
| 1 | Yes! |

---

**Route** Table, Number Type, Table, select an available type, Custom, Use Currency Symbol.

---

**Example** To display a currency symbol with numbers, the command is:

```
TableNumberUseCurrency (State: Yes!)
```

---

**See Also** ♦ TableNumberFormat

---

## TablePosition

---

**Purpose** TablePosition specifies the position of a table.

---

**Syntax** TablePosition (PositionVal: *Numeric Equivalent or Enumerated Type*; SetAbsPos: *Measurement Expression*)

---

**Parameters**

*PositionVal* Numeric Equivalent or Enumerated Type: Specifies the current table position on the page.

- 0 AlignLeft!
- 1 AlignRight!
- 2 AlignCenter!
- 3 AlignFull!
- 4 AlignAbsolute!

*SetAbsPos* Measurement Expression: Specifies an absolute position for a table from the left edge of the paper when AlignAbsolute is specified as the PositionVal parameter.

---

**Route** Table, Format, Table, select a table position.

---

**Example** To position a table 1.4 inches from the left edge of the paper, the command is:

TablePosition (PositionVal: AlignAbsolute!; SetAbsPos: 1.4")

---

---

## TableQuickSum

---

**Purpose** TableQuickSum calculates the sum of cells above the current cell in the current column. If the cell immediately above the current cell is blank, TableQuickSum calculates the sum of cells to the immediate left of the current cell. Cells above or to the left of a blank cell are not included in the calculation.

---

**Syntax** TableQuickSum ()

---

**Route** Table, Sum.

---

---

## TableRetrieve

---

**Purpose** TableRetrieve retrieves the last item moved or copied.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

|                   |                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Syntax</b>     | TableRetrieve (Reference: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                                   |
| <b>Parameters</b> |                                                                                                                                                                                                                                                   |
| <i>Reference</i>  | Numeric Equivalent or Enumerated Type: Specifies whether to retrieve a cell, block, row, or column and specifies how to insert it. <ul style="list-style-type: none"> <li>1 Block!</li> <li>2 Row!</li> <li>3 Column!</li> <li>4 Cell!</li> </ul> |
| <b>Example</b>    | To add a row and retrieve a copied row into it, the command is: <p style="text-align: center;">TableRetrieve (Reference: Row!)</p>                                                                                                                |
| <b>See Also</b>   | ♦ TableCopy ♦ TableMove                                                                                                                                                                                                                           |

---

## TableRowHeight

|                   |                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableRowHeight specifies whether row height is automatic or fixed and, if fixed, specifies the height.                                                                              |
| <b>Syntax</b>     | TableRowHeight (RowHeight: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> , Amount: <i>Measurement Expression</i> )                                                            |
| <b>Parameters</b> |                                                                                                                                                                                     |
| <i>RowHeight</i>  | Numeric Equivalent or Enumerated Type: Specifies row height as automatic (based on font size) or fixed. <ul style="list-style-type: none"> <li>0 Auto!</li> <li>1 Fixed!</li> </ul> |
| <i>Amount</i>     | Measurement Expression: Specifies row height when Fixed! is specified as the Type parameter.                                                                                        |
| <b>Route</b>      | Table, Format, Row, select Auto or Fixed.                                                                                                                                           |
| <b>Example</b>    | To specify a fixed row height of 0.25 inches, the command is: <p style="text-align: center;">TableRowHeight (RowHeight: Fixed!; Amount: 0.25")</p>                                  |
| <b>See Also</b>   | ♦ TableColumnWidth                                                                                                                                                                  |

---

## TableRowMarginBottom

---

|                   |                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableRowMarginBottom specifies the amount of space between the bottom of each cell and text in the cell.        |
| <b>Syntax</b>     | TableRowMarginBottom (Margin: <i>Measurement Expression</i> )                                                   |
| <b>Parameters</b> |                                                                                                                 |
| <i>Margin</i>     | Measurement Expression: Specifies an amount of space between the bottom of each cell and text in the cell.      |
| <b>Route</b>      | Table, Format, Row, Bottom, specify a measurement.                                                              |
| <b>Example</b>    | To specify a bottom margin of 0.05 inches, the command is:<br><code>TableRowMarginBottom (Margin: 0.05")</code> |
| <b>See Also</b>   | ♦ TableRowMarginTop                                                                                             |

---

## TableRowMarginTop

---

|                   |                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableRowMarginTop specifies an amount of space between the top of each cell and text in the cell.         |
| <b>Syntax</b>     | TableRowMarginTop (Margin: <i>Measurement Expression</i> )                                                |
| <b>Parameters</b> |                                                                                                           |
| <i>Margin</i>     | Measurement Expression: Specifies an amount of space between the top of each cell and text in the cell.   |
| <b>Route</b>      | Table, Format, Row, Top, specify a measurement.                                                           |
| <b>Example</b>    | To specify a top margin of 0.05 inches, the command is:<br><code>TableRowMarginTop (Margin: 0.05")</code> |
| <b>See Also</b>   | ♦ TableRowMarginBottom                                                                                    |



---

## TableRowNumberOfLines

---

**Purpose** TableRowNumberOfLines specifies whether multiple lines are allowed in the cells in a row.

**Syntax** TableRowNumberOfLines (RowLines: *Numeric Equivalent* or *Enumerated Type*)

**Parameters**

*RowLines* Numeric Equivalent or Enumerated Type: Specifies whether to allow multiple lines or only one line.

0 Multiple!

1 Single!

**Route** Table, Format, Row, select Single Lines or Multi Line.

**Example** To allow multiple lines in a cell, the command is:

TableRowNumberOfLines (RowLines: Multiple!)

---

## TableSelectOff

---

**Purpose** TableSelectOff turns off Select.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** TableSelectOff ()

---

## TableSelectOn

---

**Purpose** TableSelectOn selects a row, column, cell, or table.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.  
IMPORTANT: The insertion point must be in a table.

**Syntax** TableSelectOn (SelectionMode: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

|                      |                                       |
|----------------------|---------------------------------------|
| <i>SelectionMode</i> | Numeric Equivalent or Enumerated Type |
|                      | 2 Row!                                |
|                      | 3 Column!                             |
|                      | 4 Cell!                               |
|                      | 5 Table!                              |

---

**Example**

To select a row, the command is:

TableSelectOn (SelectionMode: Row!)

---

## TableSplit

---

**Purpose**

TableSplit splits a table into two tables. The second table begins with the current row.

---

**Syntax**

TableSplit ()

---

**Route**

Table, Split, Table.

---

**See Also**

♦ TableJoin

---

## TableSplitCellsDlg

---

**Purpose**

TableSplitCellsDlg displays the Split Cell dialog box.

---

**Syntax**

TableSplitCellsDlg ()

---

**Route**

Table, Split, Cell.

---

## TableSplitColumn

---

**Purpose**

TableSplitColumn splits the current cell into multiple columns.

---

**Syntax**

TableSplitColumn (ColumnCount: *Numeric Expression*)

|                    |                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------|
| <b>Parameters</b>  |                                                                                                                               |
| <i>ColumnCount</i> | Numeric Expression: Specifies the number of columns to split a cell into.                                                     |
| <b>Route</b>       | Table, Split, Cell, Columns, specify the number of columns.                                                                   |
| <b>Example</b>     | To split a cell into two columns, the command is:<br><div style="text-align: center;">TableSplitColumn (ColumnCount: 2)</div> |
| <b>See Also</b>    | ♦ TableSplitRow                                                                                                               |

---

## TableSplitRow

|                   |                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TableSplitRow splits the current cell into multiple rows.                                                            |
| <b>Syntax</b>     | TableSplitRow (RowCount: <i>Numeric Expression</i> )                                                                 |
| <b>Parameters</b> |                                                                                                                      |
| <i>RowCount</i>   | Numeric Expression: Specifies the number of rows to split a cell into.                                               |
| <b>Route</b>      | Table, Split, Cell, Rows, specify the number of rows.                                                                |
| <b>Example</b>    | To split a cell into two rows, the command is:<br><div style="text-align: center;">TableSplitRow (RowCount: 2)</div> |
| <b>See Also</b>   | ♦ TableSplitColumn                                                                                                   |

---

## TableTextToFormula

|                 |                                                                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | TableTextToFormula converts the contents of the current cell into a formula.<br><b>IMPORTANT:</b> This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | TableTextToFormula ()                                                                                                                                                                   |
| <b>See Also</b> | ♦ TableFormula                                                                                                                                                                          |

---

## TableUndelete

|                 |                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | TableUndelete undeletes the last deleted item while in table editing mode.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | TableUndelete ()                                                                                                                                                               |
| <b>See Also</b> | ♦ TableDeleteBlock ♦ TableDeleteColumn ♦ TableDeleteRow                                                                                                                        |

---

## TableUseDefaultLinesColor

|                 |                                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | TableUseDefaultLinesColor specifies the Line Style line color as the default line color for a table.<br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>   | TableUseDefaultLinesColor ()                                                                                                                                                                             |
| <b>See Also</b> | ♦ TableDefaultLineColor                                                                                                                                                                                  |

---

## TabRight

|                   |                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TabRight inserts a right aligned tab.                                                                          |
| <b>Syntax</b>     | TabRight (Type: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                          |
| <b>Parameters</b> | Numeric Equivalent or Enumerated Type: Specifies whether to include a dot leader.<br>0 Normal!<br>1 DotLeader! |
| <b>Route</b>      | Layout, Line, Other Codes, select Right under Hard Tab Codes or under Hard Tab Codes with Dot Leaders.         |
| <b>Example</b>    | To insert a right tab with a dot leader, the command is:<br>TabRight (Type: DotLeader!)                        |

---

**See Also** ♦ TabCenter ♦ TabDecimal ♦ TabLeft

---

## TabSet

---

**Purpose** TabSet specifies tab settings.

---

**Syntax** TabSet (Origin: *Numeric Equivalent* or *Enumerated Type*; {Position: *Measurement Expression*; TabType: *Numeric Equivalent* or *Enumerated Type*})

---

### Parameters

*Origin* Numeric Equivalent or Enumerated Type: Specifies whether to set tabs relative to the left edge of the paper (absolute) or the left margin setting (relative).

- 0 Absolute!
- 1 Relative!

*Position* Measurement Expression: Specifies a tab position.

*TabType* Numeric Equivalent or Enumerated Type: Specifies a tab type.

- 0 TabLeft!
- 1 TabCenter!
- 2 TabRight!
- 3 TabDecimal!
- 16 TabLeftDot!
- 17 TabCenterDot!
- 18 TabRightDot!
- 19 TabDecimalDot!

---

**Route** Layout, Line, Tab Set, Absolute or Relative, specify tab settings.

---

**Example** To set an absolute left tab at 3 inches, the command is:

```
TabSet (Origin: Absolute!; {Position: 3"; TabType: TabLeft!})
```

---

**See Also** ♦ TabAdd ♦ ?TabSettings

---

## TabSetDlg

---

**Purpose** TabSetDlg displays the Tab Set dialog box.

---

|                 |                        |
|-----------------|------------------------|
| <b>Syntax</b>   | TabSetDlg ()           |
| <b>Route</b>    | Layout, Line, Tab Set. |
| <b>See Also</b> | ♦ TabSet               |

---

---

## TemplateCopyObjectDlg

---

|                |                                                                             |
|----------------|-----------------------------------------------------------------------------|
| <b>Purpose</b> | TemplateCopyObjectDlg displays the Copy/Remove Template Objects dialog box. |
| <b>Syntax</b>  | TemplateCopyObjectDlg ()                                                    |
| <b>Route</b>   | File, Template, Options button, Edit Template, Copy/Remove Object.          |

---

---

## TemplateCreate

---

|                   |                                                                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TemplateCreate creates a new template.                                                                                                                                             |
| <b>Syntax</b>     | TemplateCreate (Filename: <i>Character Expression</i> ; BasedOn: <i>Character Expression</i> )                                                                                     |
| <b>Parameters</b> |                                                                                                                                                                                    |
| <i>Filename</i>   | Character Expression: Specifies a new template filename.                                                                                                                           |
| <i>BasedOn</i>    | Character Expression: Specifies the template filename on which a new template is based.                                                                                            |
| <b>Route</b>      | File, Template, Options button, Create Template, select options.                                                                                                                   |
| <b>Example</b>    | To create a new template file named NEWTEMP.WPT, based on the template file STANDARD.WPT, the command is:<br><br>TemplateCreate (Filename: "NEWTEMP.WPT"; BasedOn: "STANDARD.WPT") |

---

---

## TemplateCreateDlg

---

|                |                                                                                                                                                                             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | TemplateCreateDlg displays the Create Document Template dialog box.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

---

|               |                      |
|---------------|----------------------|
| <b>Syntax</b> | TemplateCreateDlg () |
|---------------|----------------------|

---

## TemplateDescriptionDlg

---

|                |                                                                      |
|----------------|----------------------------------------------------------------------|
| <b>Purpose</b> | TemplateDescriptionDlg displays the Template Description dialog box. |
| <b>Syntax</b>  | TemplateDescriptionDlg ()                                            |
| <b>Route</b>   | File, Template, Options button, Edit Template, Description.          |

---

## TemplateDlg

---

|                |                                                |
|----------------|------------------------------------------------|
| <b>Purpose</b> | TemplateDlg displays the Templates dialog box. |
| <b>Syntax</b>  | TemplateDlg ()                                 |
| <b>Route</b>   | File, Template.                                |

---

## TemplateEdit

---

|                   |                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | TemplateEdit edits a selected template file.                                                         |
| <b>Syntax</b>     | TemplateEdit (Filename: <i>Character Expression</i> )                                                |
| <b>Parameters</b> |                                                                                                      |
| <i>Filename</i>   | Character Expression: Specifies the name of a file to edit.                                          |
| <b>Route</b>      | File, Template, select Template, Options button, Edit Template.                                      |
| <b>Example</b>    | To edit a template file named NEWTEMP.WPT, the command is:<br>TemplateEdit (Filename: "NEWTEMP.WPT") |

---

---

# TemplateSelect

---

**Purpose** TemplateSelect selects a document template to use.

---

**Syntax** TemplateSelect (Filename: *Character Expression*)

---

## Parameters

*Filename* Character Expression: Specifies a template filename.

---

**Route** File, Template, select template.

---

**Example** To select the standard template file, the command is:

TemplateSelect (Filename: "STANDARD.WPT")

---

# TextBorderCreate

---

**Purpose** TextBorderCreate creates a text border.  
IMPORTANT: This command must be used in conjunction with TextBorderEnd.

---

**Syntax** TextBorderCreate (Style: *Numeric Equivalent* or *Enumerated Type*, FillName: *Numeric Equivalent* or *Enumerated Type*, BorderFunction: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*Style* Numeric Equivalent or Enumerated Type: Specifies a border style.

- 0 SpacingOnly!
- 1 SingleBorder!
- 2 DoubleBorder!
- 3 DashedBorder!
- 4 DottedBorder!
- 5 ThickBorder!
- 6 ExtraThickBorder!
- 7 ThinThickBorder!
- 8 ThickThinBorder!
- 9 ThickTopBottomBorder!
- 10 ButtonBorder!
- 11 ColumnBorderBetween!
- 12 ColumnBorderAll!
- 127 NoBorder!



|                 |                                                               |
|-----------------|---------------------------------------------------------------|
| <i>FillName</i> | Numeric Equivalent or Enumerated Type: Specifies a fill type. |
|                 | 0 Fill10!                                                     |
|                 | 1 Fill20!                                                     |
|                 | 2 Fill30!                                                     |
|                 | 3 Fill40!                                                     |
|                 | 4 Fill50!                                                     |
|                 | 5 Fill60!                                                     |
|                 | 6 Fill70!                                                     |
|                 | 7 Fill80!                                                     |
|                 | 8 Fill90!                                                     |
|                 | 9 Fill100!                                                    |
|                 | 10 FillButton!                                                |
|                 | 127 NoFill!                                                   |

|                       |                                                                 |
|-----------------------|-----------------------------------------------------------------|
| <i>BorderFunction</i> | Numeric Equivalent or Enumerated Type: Specifies a border type. |
|                       | 0 PageBorder!                                                   |
|                       | 1 ParagraphBorder!                                              |
|                       | 2 ColumnBorder!                                                 |

---

**Route** Layout, select Page, Paragraph, or Columns, Border/Fill.

---

**Example** To create a paragraph border with a single line and a 10% fill, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: Fill10!; BorderFunction: ParagraphBorder!)
TextBorderEnd (State: Save!)
```

---

**See Also** ♦ TextBorderEdit ♦ TextBorderEnd

---

## TextBorderEdit

---

**Purpose** TextBorderEdit initiates editing changes to the current border.

**IMPORTANT:** If the insertion point is not in text that has the specified border, an error is returned. This command must be used in conjunction with TextBorderEnd.

---

**Syntax** TextBorderEdit (BorderType: *Numeric Equivalent or Enumerated Type*)

---

**Parameters***BorderType*

Numeric Equivalent or Enumerated Type: Specifies a border type to edit.

- 0 Page!
- 1 Paragraph!
- 2 Column!

---

**Route**

Layout, select Page, Paragraph, or Columns, Border/Fill.

---

**Example**

To change the fill style of the current page border to 10%, the command sequence is:

```
TextBorderEdit (BorderType: Page!)
TextBorderFillStyle (FillName: Fill10!)
TextBorderEnd (State: Save!)
```

---

**See Also**

♦ TextBorderCreate ♦ TextBorderEnd

---

## TextBorderEnd

---

**Purpose**

TextBorderEnd ends border creation or editing and specifies whether to save changes. IMPORTANT: This command must be preceded by TextBorderCreate or TextBorderEdit.

---

**Syntax**TextBorderEnd (State: *Numeric Equivalent or Enumerated Type*)

---

**Parameters***State*

Numeric Equivalent or Enumerated Type: Specifies whether to save changes to a border.

- 0 Cancel!
- 1 Save!

---

**Route**

Layout, select Page, Paragraph, or Columns, Border/Fill.

---

**Example**

To create a paragraph border with a single line and a 10% fill, the command sequence is:

```
TextBorderCreate (Style: SingleBorder!; FillName: Fill10!; BorderFunction: ParagraphBorder!)
TextBorderEnd (State: Save!)
```

---

**See Also**

♦ TextBorderCreate ♦ TextBorderEdit

---

# TextBorderFillStyle

---

## Purpose

TextBorderFillStyle specifies a fill style for the current border.

**IMPORTANT:** This command is not recordable. To use this command, you must type it into the macro.

**IMPORTANT:** If the insertion point is not in text with a border, an error is returned. This command must be used in conjunction with TextBorderEdit and TextBorderEnd.

---

## Syntax

TextBorderFillStyle (FillName: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

### *FillName*

Numeric Equivalent or Enumerated Type: Specifies a fill style.

- 0 Fill10!
  - 1 Fill20!
  - 2 Fill30!
  - 3 Fill40!
  - 4 Fill50!
  - 5 Fill60!
  - 6 Fill70!
  - 7 Fill80!
  - 8 Fill90!
  - 9 Fill100!
  - 10 FillButton!
  - 127 NoFill!
- 

## Example

To change the fill style of the current page border to 10%, the command sequence is:

```
TextBorderEdit (BorderType: Page!)
TextBorderFillStyle (FillName: Fill10!)
TextBorderEnd (State: Save!)
```

---

## See Also

- ♦ TextBorderEdit
-

---

# TextBorderStyle

---

## Purpose

TextBorderStyle specifies a border style for the current border.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

IMPORTANT: If the insertion point is not in text with a border, an error is returned. This command must be used in conjunction with TextBorderEdit and TextBorderEnd.

---

## Syntax

TextBorderStyle (Style: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*Style*

Numeric Equivalent or Enumerated Type: Specifies a border style.

- 0 SpacingOnly!
  - 1 SingleBorder!
  - 2 DoubleBorder!
  - 3 DashedBorder!
  - 4 DottedBorder!
  - 5 ThickBorder!
  - 6 ExtraThickBorder!
  - 7 ThinThickBorder!
  - 8 ThickThinBorder!
  - 9 ThickTopBottomBorder!
  - 10 ButtonBorder!
  - 11 ColumnBorderBetween!
  - 12 ColumnBorderAll!
  - 127 NoBorder!
- 

## Example

To change the border style of the current page border to thick, the command sequence is:

```
TextBorderEdit (BorderType: Page!)
TextBorderStyle (Style: Thick!)
TextBorderEnd (State: Save!)
```

---

## See Also

♦ TextBorderEdit ♦ TextBorderEnd

---

---

# TextBoxCreate

---

**Purpose** TextBoxCreate creates a text box and opens the Text Box Editor.

---

**Syntax** TextBoxCreate ()

---

**Route** Graphics, Text Box.

---

**See Also** ♦ SubstructureExit

---

# TextColor

---

**Purpose** TextColor specifies a text color.

---

**Syntax** TextColor (ColorName: *Character Expression*; Red: *Numeric Expression*; Green: *Numeric Expression*; Blue: *Numeric Expression*)

---

## Parameters

*ColorName* Character Expression: Specifies a predefined color name.

*Red* Numeric Expression: Specifies the amount of red in a color. Values range from 0 to 255.

*Green* Numeric Expression: Specifies the amount of green in a color. Values range from 0 to 255.

*Blue* Numeric Expression: Specifies the amount of blue in a color. Values range from 0 to 255.

---

**Route** Layout, Font, Color button, select a color.

---

**Example** To specify red text, the command is:

TextColor (ColorName: "Red"; Red: 255; Green: 0; Blue: 0)

---

**See Also** ♦ TextShade ♦ ?TextColorBlue ♦ ?TextColorGreen ♦ ?TextColorRed ♦ ?TextShade

---

---

## TextShade

**Purpose** TextShade specifies the intensity of a text color, where 100% is full intensity.

**Syntax** TextShade (ShadingValue: *Numeric Expression*)

### Parameters

*ShadingValue* Numeric Expression: Specifies a shading percentage for a text color.

**Route** Layout, Font, specify a shading percentage under Color Options.

**Example** To specify a blue text with half intensity, the command sequence is:

```
TextColor (ColorName: "Blue"; Red: 0; Green: 0; Blue: 255)
TextShade (ShadingValue: 50)
```

**See Also** ♦ TextColor

---

## ThousandsSeparator

**Purpose** ThousandsSeparator specifies a character that appears in numbers greater than or equal to 1000 to separate each three digits.

IMPORTANT: Use with the Comma! number type specified in TableCellNumberFormat, TableColumnNumberFormat, or TableNumberFormat, or use with TableCellNumberUseCommas, TableColumnNumberUseCommas, or TableNumberUseCommas.

**Syntax** ThousandsSeparator (Character: *Character Expression*)

### Parameters

*Character* Character Expression: Specifies a character that separates every three digits.

**Route** Layout, Line, Other Codes, Thousands Separator.

**Example** To specify a period (.) as the thousands separator, the command is:

```
ThousandsSeparator (Character: ".")
```

**See Also** ♦ TableCellNumberFormat ♦ TableCellNumberUseCommas  
♦ TableColumnNumberFormat ♦ TableColumnNumberUseCommas  
♦ TableNumberFormat ♦ TableNumberUseCommas ♦ ?ThousandsSeparatorChar

---

## ToACreateFullFormDlg

---

**Purpose** ToACreateFullFormDlg displays the Create Full Form dialog box.

**Syntax** ToACreateFullFormDlg ()

**Route** Tools, Table of Authorities, Create Full Form.

---

## ToADefineDlg

---

**Purpose** ToADefineDlg displays the Define Table of Authorities dialog box.

**Syntax** ToADefineDlg ()

**Route** Tools, Table of Authorities, Define.

---

## ToADefinition

---

**Purpose** ToADefinition specifies a section and inserts a table of authorities definition.

**Syntax** ToADefinition (SectionName: *Character Expression*; Style: *Numeric Equivalent* or *Enumerated Type*; Numbering: *Numeric Equivalent* or *Enumerated Type*; Underline: *Numeric Equivalent* or *Enumerated Type*; PageCombine: *Numeric Equivalent* or *Enumerated Type*; NumberingFormat: *Character Expression*)

---

### Parameters

*SectionName* Character Expression: Specifies a section name.

*Style* Numeric Equivalent or Enumerated Type: Specifies a style for the current section.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!

27 ToC4Style!  
28 ToC5Style!  
29 Index1Style!  
30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!



75 Heading8Style!

|                        |                                                                                                                                                                          |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Numbering</i>       | Numeric Equivalent or Enumerated Type: Specifies how to display page references.<br>0 DefNone!<br>1 DefFollows!<br>2 DefParentheses!<br>3 DefFlushRight!<br>4 DefLeader! |
| <i>Underline</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to allow underlining.<br>0 NoToAUnderline!<br>1 ToAUnderline!                                                   |
| <i>PageCombine</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to combine sequential page numbers.<br>0 NoCombine!<br>1 Combine!                                               |
| <i>NumberingFormat</i> | Character Expression: Specifies how page numbers appear in references.                                                                                                   |

---

**Route** Tools, Table of Authorities, Define, select a section, Insert.

---

**Example** To specify a section named SWITCHS and insert a definition, the command is:

ToADefinition (SectionName: "SWITCHS"; Style: NormalStyle!; Numbering: DefLeader!; Underline: NoToAUnderline!; PageCombine: NoCombine!; Numbering Format: "Page #")

---

**See Also** • ToADefinitionCreate • ?ToADefinition

---

## ToADefinitionCombine

---

**Purpose** ToADefinitionCombine specifies whether to combine sequential page numbers in a section.

---

**Syntax** ToADefinitionCombine (SectionName: *Character Expression*; Combine: *Numeric Equivalent or Enumerated Type*)

---

### Parameters

*SectionName* Character Expression: Specifies a section name.

*Combine* Numeric Equivalent or Enumerated Type: Specifies whether to combine sequential page numbers.

0 NoCombine!  
1 Combine!

---

**Route** Tools, Table of Authorities, Define, Create or Edit, Use Dash to Show Consecutive Pages.

---

**Example** To combine sequential page numbers in a section named SWITCHS, the command is:

ToADefinitionCombine (SectionName: "SWITCHS"; Combine: Combine!)

---

## ToADefinitionCreate

---

**Purpose** ToADefinitionCreate creates a table of authorities definition.

**Syntax** ToADefinitionCreate (SectionName: *Character Expression*; Style: *Numeric Equivalent* or *Enumerated Type*; Numbering: *Numeric Equivalent* or *Enumerated Type*; Underline: *Numeric Equivalent* or *Enumerated Type*; PageCombine: *Numeric Equivalent* or *Enumerated Type*; NumberingFormat: *Character Expression*)

---

### Parameters

*SectionName* Character Expression: Specifies a section name.

*Style* Numeric Equivalent or Enumerated Type: Specifies a style for the current section.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TextBoxNumberStyle!
- 20 TextBoxNumberStyle!
- 21 UserBoxNumberStyle!
- 22 EquationNumberStyle!
- 23 HypertextStyle!
- 24 ToC1Style!
- 25 ToC2Style!
- 26 ToC3Style!
- 27 ToC4Style!
- 28 ToC5Style!
- 29 Index1Style!
- 30 Index2Style!
- 31 ListStyle!
- 32 ToAStyle!
- 33 DocStyle!
- 34 BoxTextStyle!
- 35 CaptionStyle!
- 36 FootnoteStyle!
- 37 EndnoteStyle!
- 38 CommentStyle!

39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!  
72 Heading5Style!  
73 Heading6Style!  
74 Heading7Style!  
75 Heading8Style!

## *Numbering*

Numeric Equivalent or Enumerated Type: Specifies how to display page references.

0 DefNone!  
1 DefFollows!  
2 DefParentheses!  
3 DefFlushRight!  
4 DefLeader!

|                        |                                                                                                                                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Underline</i>       | Numeric Equivalent or Enumerated Type: Specifies whether to allow underlining.<br>0 NoToAUnderline!<br>1 ToAUnderline!                                                                                                                                                                  |
| <i>PageCombine</i>     | Numeric Equivalent or Enumerated Type: Specifies whether to combine sequential page numbers.<br>0 NoCombine!<br>1 Combine!                                                                                                                                                              |
| <i>NumberingFormat</i> | Character Expression: Specifies how page numbers appear in references.                                                                                                                                                                                                                  |
| <b>Route</b>           | Tools, Table of Authorities, Define, Create, enter a name, specify numbering format and current style options.                                                                                                                                                                          |
| <b>Example</b>         | To create a definition for a section named SWITCHS with flush right page numbers and dot leaders, the command is:<br><br>ToADefinitionCreate (SectionName: "SWITCHS"; Style: ToAStyle!; Numbering: DefLeader!; Underline: ToAUnderline!; PageCombine: Combine!; NumberingFormat: "One") |
| <b>See Also</b>        | ♦ ToADefinitionDelete                                                                                                                                                                                                                                                                   |

---

## ToADefinitionDelete

|                    |                                                                                                                   |
|--------------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | ToADefinitionDelete deletes a table of authorities section definition.                                            |
| <b>Syntax</b>      | ToADefinitionDelete (SectionName: <i>Character Expression</i> )                                                   |
| <b>Parameters</b>  |                                                                                                                   |
| <i>SectionName</i> | Character Expression: Specifies a section name.                                                                   |
| <b>Route</b>       | Tools, Table of Authorities, Define, select a section, Delete.                                                    |
| <b>Example</b>     | To delete a section definition named SWITCHS, the command is:<br><br>ToADefinitionDelete (SectionName: "SWITCHS") |
| <b>See Also</b>    | ♦ ToADefinitionCreate                                                                                             |

---

## ToADefinitionMode

---

|                    |                                                                                                                                                                                                      |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | ToADefinitionMode changes the display of page references in a section.                                                                                                                               |
| <b>Syntax</b>      | ToADefinitionMode (SectionName: <i>Character Expression</i> ; Numbering: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                       |
| <b>Parameters</b>  |                                                                                                                                                                                                      |
| <i>SectionName</i> | Character Expression: Specifies a section name.                                                                                                                                                      |
| <i>Numbering</i>   | Numeric Equivalent or Enumerated Type: Specifies how to display page references.<br>0 DefNone!<br>1 DefFollows!<br>2 DefFollowsParenthesis!<br>3 DefFlushRight!<br>4 DefLeader!                      |
| <b>Route</b>       | Tools, Table of Authorities, Define, select a section, Edit, select a Position option.                                                                                                               |
| <b>Example</b>     | To change the display of page references in a section called SWITCHS to flush right with no dot leader, the command is:<br><br>ToADefinitionMode (SectionName: "SWITCHS"; Numbering: DefFlushRight!) |
| <b>See Also</b>    | ♦ ToADefinitionCreate                                                                                                                                                                                |

---

## ToADefinitionRename

---

|                   |                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ToADefinitionRename renames a section.                                                                                                           |
| <b>Syntax</b>     | ToADefinitionRename (OldSection: <i>Character Expression</i> ; NewSection: <i>Character Expression</i> )                                         |
| <b>Parameters</b> |                                                                                                                                                  |
| <i>OldSection</i> | Character Expression: Specifies a section name.                                                                                                  |
| <i>NewSection</i> | Character Expression: Specifies a new name for a section.                                                                                        |
| <b>Route</b>      | Tools, Table of Authorities, Define, select a section, Edit, Name, enter a name.                                                                 |
| <b>Example</b>    | To rename a section from SWITCHS to UTAH SWITCHS, the command is:<br><br>ToADefinitionRename (OldSection: "SWITCHS"; NewSection: "UTAH SWITCHS") |

---

**See Also**

♦ ToADefinitionCreate

---

## ToADefinitionRetrieve

**Purpose**

ToADefinitionRetrieve retrieves a section from another document into the current document.

**Syntax**

ToADefinitionRetrieve (Filename: *Character Expression*; SectionName: *Character Expression*)

**Parameters**

*Filename*

Character Expression: Specifies a filename.

*SectionName*

Character Expression: Specifies a section name.

**Route**

Tools, Table of Authorities, Define, Retrieve, enter a filename, mark sections to retrieve.

**Example**

To retrieve a section named SWITCHS in a file named PROBATE, the command is:

```
ToADefinitionRetrieve (Filename: "PROBATE"; SectionName: "SWITCHS")
```

**See Also**

♦ ToADefinitionCreate

---

## ToADefinitionStyle

**Purpose**

ToADefinitionStyle changes the style for a section.

**Syntax**

ToADefinitionStyle (SectionName: *Character Expression*; Style: *Numeric Equivalent or Enumerated Type*)

**Parameters**

*SectionName*

Character Expression: Specifies a section name.

*Style*

Numeric Equivalent or Enumerated Type: Specifies a style for the current section.

- 0 NormalStyle!
- 16 FootnoteNumberInDocumentStyle!
- 17 EndnoteNumberInDocumentStyle!
- 18 FigureNumberStyle!
- 19 TableBoxNumberStyle!
- 20 TextBoxNumberStyle!

21 UserBoxNumberStyle!  
22 EquationNumberStyle!  
23 HypertextStyle!  
24 ToC1Style!  
25 ToC2Style!  
26 ToC3Style!  
27 ToC4Style!  
28 ToC5Style!  
29 Index1Style!  
30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
35 CaptionStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Legal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!

- 69 Heading2Style!
- 70 Heading3Style!
- 71 Heading4Style!
- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

---

**Route** Tools, Table of Authorities, Define, select a section, Edit, Change, select a style, Select.

---

**Example** To specify the TableofCont1 style for a section named SWITCHS, the command is:  
 ToADefinitionStyle (SectionName: "SWITCHS"; Style: ToC1Style!)

---

**See Also** ♦ ToADefinitionCreate

---

## ToADefinitionUnderline

---

**Purpose** ToADefinitionUnderline specifies whether underlining is allowed in a table of authorities section.

---

**Syntax** ToADefinitionUnderline (Section: *Character Expression*; Action: *Numeric Equivalent* or *Enumerated Type*)

---

### Parameters

*Section* Character Expression: Specifies a table of authorities section name.

*Action* Numeric Equivalent or Enumerated Type: Specifies whether to allow underlining in the specified section.

- 0 NoToAUnderline!
- 1 ToAUnderline!

---

**Route** Tools, Table of Authorities, Define, Create or Edit, select Underlining Allowed.

---

**Example** To allow underlining in a table of authorities section named First Section, the command is:

ToADefinitionUnderline (Section: "First Section"; Action: ToAUnderline!)



---

## ToAEditFullForm

---

**Purpose** ToAEditFullForm initiates editing changes to a full form. Use SubstructureExit to end editing and save changes.

---

**Syntax** ToAEditFullForm (ShortForm: *Character Expression*)

---

### Parameters

*ShortForm* Character Expression: Specifies the short form of a full form to edit.

---

**Route** Tools, Table of Authorities, Edit Full Form, select a form.

---

**Example** To edit a full form with the short form Brown v. Board of Education, the command sequence is:

```
ToAEditFullForm (ShortForm: "Brown v. Board of Education")
Type (Text: "385 U.S.")
SubstructureExit ()
```

---

**See Also** ♦ ToAMark

---

## ToAEditFullFormCBar

---

**Purpose** ToAEditFullFormCBar displays the Table of Authorities feature bar.

---

**Syntax** ToAEditFullFormCBar ()

---

**Route** Tools, Table of Authorities.

---

## ToAEditFullFormDlg

---

**Purpose** ToAEditFullFormDlg displays the Edit Full Form dialog box.

---

**Syntax** ToAEditFullFormDlg ()

---

**Route** Tools, Table of Authorities, Edit Full Form.

---

**See Also** ♦ ToAEditFullForm

---

## ToAEditPageFormat

---

**Purpose**

ToAEditPageFormat changes the page format for a section.

**Syntax**

ToAEditPageFormat (Listname: *Character Expression*; NumberFormat: *Numeric Equivalent or Enumerated Type*)

**Parameters**

*Listname*

Character Expression: Specifies a section name.

*NumberFormat*

Numeric Equivalent or Enumerated Type: Specifies a page format. This parameter also accepts a character expression to specify a custom format.

0 DocFormat!

**Route**

Tools, Table of Authorities, Define, select a section, Edit, Page Numbering, select options.

**Example**

To include a chapter number and a period in front of the page number for a section named SWITCHS, the command is:

```
ToAEditPageFormat (Listname: "SWITCHS"; NumberFormat: "[chpt #].[page #]")
```

**See Also**

♦ ToADefinitionCreate

---

---

## ToAEditSection

---

**Purpose**

ToAEditSection specifies a table of authorities section to edit.

**Syntax**

ToAEditSection (OldName: *Character Expression*; NewName: *Character Expression*)

**Parameters**

*OldName*

Character Expression: Specifies the current section.

*NewName*

Character Expression: Specifies the section to edit.

**Route**

Tools, Table of Authorities, Edit Full Form, select a section.

**Example**

To switch from a section named First Section to edit a section named Second Section, the command is:

```
ToAEditSection (OldName: "First Section"; NewName: "Second Section")
```

---

## ToAEditShortForm

---

|                   |                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ToAEditShortForm renames a short form.                                                                                                                                                            |
| <b>Syntax</b>     | ToAEditShortForm (OldName: <i>Character Expression</i> ; NewName: <i>Character Expression</i> )                                                                                                   |
| <b>Parameters</b> |                                                                                                                                                                                                   |
| <i>OldName</i>    | Character Expression: Specifies a short form.                                                                                                                                                     |
| <i>NewName</i>    | Character Expression: Specifies a new short form.                                                                                                                                                 |
| <b>Route</b>      | Tools, Table of Authorities, Edit Full Form, select a section, OK, enter a short form.                                                                                                            |
| <b>Example</b>    | To rename a short form Brown v. Board of Education to Brown v. United States, the command is:<br><br>ToAEditShortForm (OldName: "Brown v. Board of Education"; NewName: "Brown v. United States") |
| <b>See Also</b>   | ♦ ToAMark                                                                                                                                                                                         |

---

## ToAMark

---

|                    |                                                                                                                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>     | ToAMark marks selected text as a table of authorities reference.                                                                                                                                                 |
| <b>Syntax</b>      | ToAMark (SectionName: <i>Character Expression</i> ; ShortForm: <i>Character Expression</i> )                                                                                                                     |
| <b>Parameters</b>  |                                                                                                                                                                                                                  |
| <i>SectionName</i> | Character Expression: Specifies a section name.                                                                                                                                                                  |
| <i>ShortForm</i>   | Character Expression: Specifies a short form.                                                                                                                                                                    |
| <b>Route</b>       | Tools, Table of Authorities, Create Full Form, specify a section name and a short form.                                                                                                                          |
| <b>Example</b>     | To mark a reference with the short form Brown v. Board of Education and include it in a section named SWITCHS, the command is:<br><br>ToAMark (SectionName: "SWITCHS"; ShortForm: "Brown v. Board of Education") |
| <b>See Also</b>    | ♦ ToADefinitionCreate                                                                                                                                                                                            |

---

# ToAMarkShortForm

---

|                   |                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | ToAMarkShortForm marks a short form.                                                                                                   |
| <b>Syntax</b>     | ToAMarkShortForm (ShortForm: <i>Character Expression</i> )                                                                             |
| <b>Parameters</b> |                                                                                                                                        |
| <i>ShortForm</i>  | Character Expression: Specifies a short form.                                                                                          |
| <b>Route</b>      | Tools, Table of Authorities, Mark.                                                                                                     |
| <b>Example</b>    | To mark the short form Brown v. Board of Education, the command is:<br><br>ToAMarkShortForm (ShortForm: "Brown v. Board of Education") |
| <b>See Also</b>   | ♦ ToAMark                                                                                                                              |

---

# ToCDefine

---

|                        |                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>         | ToCDefine defines a table of contents and inserts the definition.                                                                                                                                                                                                                                                                                 |
| <b>Syntax</b>          | ToCDefine (WrapLastLevel: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ,<br>NumberingFormat: <i>Character Expression</i> ; Style: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ;<br>Numbering: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                |
| <b>Parameters</b>      |                                                                                                                                                                                                                                                                                                                                                   |
| <i>WrapLastLevel</i>   | Numeric Equivalent or Enumerated Type: Specifies whether to wrap the last level.<br><br>0 No!<br>1 Yes!                                                                                                                                                                                                                                           |
| <i>NumberingFormat</i> | Character Expression: Specifies a page format.                                                                                                                                                                                                                                                                                                    |
| <i>Style</i>           | Numeric Equivalent or Enumerated Type: Specifies a style for each table of contents level.<br><br>0 NormalStyle!<br>16 FootnoteNumberInDocumentStyle!<br>17 EndnoteNumberInDocumentStyle!<br>18 FigureNumberStyle!<br>19 TextBoxNumberStyle!<br>20 TextBoxNumberStyle!<br>21 UserBoxNumberStyle!<br>22 EquationNumberStyle!<br>23 HypertextStyle! |

24 ToC1Style!  
25 ToC2Style!  
26 ToC3Style!  
27 ToC4Style!  
28 ToC5Style!  
29 Index1Style!  
30 Index2Style!  
31 ListStyle!  
32 ToAStyle!  
33 DocStyle!  
34 BoxTextStyle!  
36 FootnoteStyle!  
37 EndnoteStyle!  
38 CommentStyle!  
39 HeaderAStyle!  
40 HeaderBStyle!  
41 FooterAStyle!  
42 FooterBStyle!  
43 WatermarkAStyle!  
44 WatermarkBStyle!  
45 ParagraphOutlnStyle!  
46 CaptionStyle!  
46 OutlineStyle!  
47 LegalOutlnStyle!  
48 BulletsOutlnStyle!  
49 HeadingsOutlnStyle!  
50 Lcgal\_2OutlnStyle!  
51 NumbersOutlnStyle!  
52 Level1Style!  
53 Level2Style!  
54 Level3Style!  
55 Level4Style!  
56 Level5Style!  
57 Level6Style!  
58 Level7Style!  
59 Level8Style!  
60 Legal1Style!  
61 Legal2Style!  
62 Legal3Style!  
63 Legal4Style!  
64 Legal5Style!  
65 Legal6Style!  
66 Legal7Style!  
67 Legal8Style!  
68 Heading1Style!  
69 Heading2Style!  
70 Heading3Style!  
71 Heading4Style!

- 72 Heading5Style!
- 73 Heading6Style!
- 74 Heading7Style!
- 75 Heading8Style!

*Numbering*

Numeric Equivalent or Enumerated Type: Specifies the format for references.

- 0 DefNone!
- 1 DefFollows!
- 2 DefParentheses!
- 3 DefFlushRight!
- 4 DefLeader!

---

**Route** Tools, Table of Contents, Define, specify a number of levels, specify and select options.

---

**Example** To create a single-level table of contents with flush right dot leader page numbers, the command is:

```
ToCDefine (WrapLastLevel: LevelWrap! NumberingFormat: "Page #"; {Style: NormalStyle!; Numbering: DefLeader!})
```

---

**See Also** ♦ ToCMark ♦ ?ToCDefinition

---

## ToCDefineDlg

---

**Purpose** ToCDefineDlg displays the Define Table of Contents dialog box.

---

**Syntax** ToCDefineDlg ()

---

**Route** Tools, Table of Contents, Define.

---

**See Also** ♦ ToCMark

---

## ToCMark

---

**Purpose** ToCMark marks selected text for inclusion in a table of contents.

---

**Syntax** ToCMark (Level: *Numeric Expression*)

---

## Parameters

*Level* Numeric Expression: Specifies a table of contents level.

**Route** Tools, Table of Contents, Mark, specify a level.

**Example** To mark text for level 1 of a table of contents, the command is:

ToCMark (Level: 1)

**See Also** ♦ ToCDefine

---

## Type

**Purpose** Type inserts text at the insertion point.

**Syntax** Type (Text: *Character Expression*)

### Parameters

*Text* Character Expression: Specifies text to enter at the insertion point.

**Route** Use the keyboard to type in a document window.

**Example** To type “WordPerfect 6.0 for Windows” in a document, the command is:

Type (Text: “WordPerfect 6.0 for Windows”)

**See Also** ♦ TypeChar

---

## TypeChar

**Purpose** TypeChar inserts a character from the WordPerfect character sets.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

**Syntax** TypeChar (CharacterSet: *Numeric Expression*; Offset: *Numeric Expression*)

---

---

## Parameters

*CharacterSet*

Numeric Expression: Specifies a WordPerfect character set.

*Offset*

Numeric Expression: Specifies a character in a WordPerfect character set.

---

## Example

To insert the section symbol (§), which is character 6 of set 4, the command is:

TypeChar (CharacterSet: 4; Offset: 6)

---

## See Also

- ♦ Type

---

# Typeover

---

## Purpose

Typeover specifies whether Typeover mode is on.

---

## Syntax

Typeover (State: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*State*

Numeric Equivalent or Enumerated Type: Specifies whether Typeover is on.

0 Off!

1 On!

---

## Route

Press Insert.

---

## Example

To turn on Typeover, the command is:

Typeover (State: On!)

---

## See Also

- ♦ ?TypeoverActive

---

# TypeoverKey

---

## Purpose

TypeoverKey toggles Typeover.

IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. For a recordable command with similar functionality, use Typeover.

---

## Syntax

TypeoverKey ()



---

**See Also** ♦ Typeover

---

## TypesetManualKerningDlg

---

**Purpose** TypesetManualKerningDlg displays the Manual Kerning dialog box.  
IMPORTANT: This command is not recordable. To use this command, you must type it into the macro.

---

**Syntax** TypesetManualKerningDlg ()

---

**See Also** ♦ KerningSpacing

---

## Undelete

---

**Purpose** Undelete restores up to the last three deletions.

---

**Syntax** Undelete (Which: *Numeric Expression*)

---

**Parameters**  
*Which* Numeric Expression: Specifies a level to undelete. Values range from 1 to 3.

---

**Route** Edit, Undelete, Restore.

---

**Example** To cancel the last deletion, the command is:  
Undelete (Which: 1)

---

**See Also** ♦ Undo

---

## UndeleteDlg

---

**Purpose** UndeleteDlg displays the Undelete dialog box.

---

**Syntax** UndeleteDlg ()

---

|              |                 |
|--------------|-----------------|
| <b>Route</b> | Edit, Undelete. |
|--------------|-----------------|

---

## UnderlineSpaces

---

|                |                                                        |
|----------------|--------------------------------------------------------|
| <b>Purpose</b> | UnderlineSpaces specifies whether to underline spaces. |
|----------------|--------------------------------------------------------|

---

|               |                                                                               |
|---------------|-------------------------------------------------------------------------------|
| <b>Syntax</b> | UnderlineSpaces (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|-------------------------------------------------------------------------------|

---

### Parameters

|              |                                                                                                  |
|--------------|--------------------------------------------------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether to underline spaces.<br>0 No!<br>1 Yes! |
|--------------|--------------------------------------------------------------------------------------------------|

---

|              |                       |
|--------------|-----------------------|
| <b>Route</b> | Layout, Font, Spaces. |
|--------------|-----------------------|

---

|                |                                                                                     |
|----------------|-------------------------------------------------------------------------------------|
| <b>Example</b> | To underline spaces in a document, the command is:<br>UnderlineSpaces (State: Yes!) |
|----------------|-------------------------------------------------------------------------------------|

---

|                 |                                    |
|-----------------|------------------------------------|
| <b>See Also</b> | ♦ UnderlineTabs ♦ ?UnderlineSpaces |
|-----------------|------------------------------------|

---

## UnderlineTabs

---

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | UnderlineTabs specifies whether to underline tabs. |
|----------------|----------------------------------------------------|

---

|               |                                                                             |
|---------------|-----------------------------------------------------------------------------|
| <b>Syntax</b> | UnderlineTabs (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> ) |
|---------------|-----------------------------------------------------------------------------|

---

### Parameters

|              |                                                                                                |
|--------------|------------------------------------------------------------------------------------------------|
| <i>State</i> | Numeric Equivalent or Enumerated Type: Specifies whether to underline tabs.<br>0 No!<br>1 Yes! |
|--------------|------------------------------------------------------------------------------------------------|

---

|              |                     |
|--------------|---------------------|
| <b>Route</b> | Layout, Font, Tabs. |
|--------------|---------------------|

---

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| <b>Example</b> | To underline tabs, the command is:<br>UnderlineTabs (State: Yes!) |
|----------------|-------------------------------------------------------------------|

---

**See Also** ♦ UnderlineSpaces ♦ ?UnderlineTabs

---

## Undo

---

**Purpose** Undo restores the last document change. Choosing Undo a second time restores the last undo.

---

**Syntax** Undo ()

---

**Route** Edit, Undo.

---

**See Also** ♦ Undelete

---

## UserFunction

---

**Purpose** UserFunction executes a third-party application command.

---

**Syntax** UserFunction (Action: *Character Expression*; Data: *Character Expression*)

### Parameters

*Action* Character Expression: Specifies an action a third-party program is to perform, and has the following form: <signature>:<action>. The <signature> has four characters and identifies a third-party application. The <action> specifies a third-party command.

*Data* Character Expression: Specifies data necessary for Action.

---

## ViewDraft

---

**Purpose** ViewDraft displays a document without features such as Headers, Footers, and Footnotes.

---

**Syntax** ViewDraft ()

---

**Route** View, Draft.

---

---

**See Also** ♦ [ViewPage](#) ♦ [ViewTwoPage](#)

---

## ViewOutline

---

**Purpose** ViewOutline displays the Outline feature bar.

---

**Syntax** ViewOutline ()

---

**Route** Tools, Outline.

---

## ViewPage

---

**Purpose** ViewPage displays a document as it prints with features such as Headers, Footers, and Footnotes.

---

**Syntax** ViewPage ()

---

**Route** View, Page.

---

**See Also** ♦ [ViewDraft](#) ♦ [ViewTwoPage](#)

---

## ViewTwoPage

---

**Purpose** ViewTwoPage displays two document pages.

---

**Syntax** ViewTwoPage ()

---

**Route** View, Two Page.

---

**See Also** ♦ [ViewDraft](#) ♦ [ViewPage](#)

---

## VLineCreate

|                |                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------|
| <b>Purpose</b> | VLineCreate creates a vertical line at the insertion point between the top and bottom margin. |
| <b>Syntax</b>  | VLineCreate ()                                                                                |
| <b>Route</b>   | Graphics, Vertical Line.                                                                      |

---

## VolumeNumber

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | VolumeNumber specifies the current volume number.                                               |
| <b>Syntax</b>     | VolumeNumber (Volume: <i>Numeric Expression</i> )                                               |
| <b>Parameters</b> |                                                                                                 |
| <i>Volume</i>     | Numeric Expression: Specifies a volume number.                                                  |
| <b>Route</b>      | Layout, Page, Numbering, Value, New Volume Number.                                              |
| <b>Example</b>    | To specify a current page as volume 5, the command is:<br><code>VolumeNumber (Volume: 5)</code> |
| <b>See Also</b>   | ♦ ?VolumeNumber                                                                                 |

---

## VolumeNumberDecrement

|                |                                                                                            |
|----------------|--------------------------------------------------------------------------------------------|
| <b>Purpose</b> | VolumeNumberDecrement decreases the current volume number by one.                          |
| <b>Syntax</b>  | VolumeNumberDecrement ()                                                                   |
| <b>Route</b>   | Layout, Page, Numbering, Value, lower setting in Increase/Decrease Existing Volume Number. |

---

## VolumeNumberDisplay

---

|                |                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | VolumeNumberDisplay displays a current volume number.                                                          |
| <b>Syntax</b>  | VolumeNumberDisplay ()                                                                                         |
| <b>Route</b>   | Layout, Page, Numbering, Value, select Insert and Display at Insertion Point in the Volume Settings group box. |

---

## VolumeNumberIncrement

---

|                |                                                                         |
|----------------|-------------------------------------------------------------------------|
| <b>Purpose</b> | VolumeNumberIncrement increases a current volume number by one.         |
| <b>Syntax</b>  | VolumeNumberIncrement ()                                                |
| <b>Route</b>   | Layout, Page, Numbering, Value, increase Existing Volume Number by one. |

---

## VolumeNumberMethod

---

|                        |                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>         | VolumeNumberMethod specifies a numbering type to display volume numbers.                                                                                                                                                                               |
| <b>Syntax</b>          | VolumeNumberMethod (NumberingMethod: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                                                                                                                                                             |
| <b>Parameters</b>      |                                                                                                                                                                                                                                                        |
| <i>NumberingMethod</i> | Numeric Equivalent or Enumerated Type: Specifies a numbering type to display volume numbers.<br><ul style="list-style-type: none"><li>0 Numbers!</li><li>1 LowerLetters!</li><li>2 UpperLetters!</li><li>3 LowerRoman!</li><li>4 UpperRoman!</li></ul> |
| <b>Route</b>           | Layout, Page, Numbering, Options, Volume, select a type.                                                                                                                                                                                               |
| <b>Example</b>         | To display volume numbers as uppercase roman numerals, the command is:<br><pre>VolumeNumberMethod (NumberingMethod: UpperRoman!)</pre>                                                                                                                 |

---

---

**See Also**

♦ ?VolumeNumberMethod

---

## WatermarkA

---

**Purpose**

WatermarkA creates, edits, or turns off Watermark A.

---

**Syntax**

WatermarkA (Action: *Numeric Equivalent* or *Enumerated Type*, Occurrence: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Action* Numeric Equivalent or Enumerated Type: Specifies the state of Watermark A.

- 0 Off!
- 1 Create!
- 2 Edit!

*Occurrence* Numeric Equivalent or Enumerated Type: Specifies Watermark pages.

- 0 NoPages!
- 1 OddPages!
- 2 EvenPages!
- 3 AllPages!

---

**Route**

Layout, Watermark, Watermark A, Create or Edit or Discontinue.

---

**Example**

To create Watermark A on all pages, the command is:

WatermarkA (Action: Create!; Occurrence: AllPages!)

---

**See Also**

♦ WatermarkB ♦ ?WatermarkAOccur

---

## WatermarkB

---

**Purpose**

WatermarkB creates, edits, or turns off Watermark B.

---

**Syntax**

WatermarkB (Action: *Numeric Equivalent* or *Enumerated Type*, Occurrence: *Numeric Equivalent* or *Enumerated Type*)

---

**Parameters**

*Action* Numeric Equivalent or Enumerated Type: Specifies the state of Watermark B.

- 0 Off!
- 1 Create!
- 2 Edit!

*Occurrence* Numeric Equivalent or Enumerated Type: Specifies Watermark pages.

- 0 NoPages!
- 1 OddPages!
- 2 EvenPages!
- 3 AllPages!

---

**Route** Layout, Watermark, Watermark B, Create or Edit or Discontinue.

---

**Example** To create Watermark B on all pages, the command is:

WatermarkB (Action: Create!; Occurrence: AllPages!)

---

**See Also** ♦ WatermarkA ♦ ?WatermarkBOccur

---

## WatermarkDlg

---

**Purpose** WatermarkDlg displays the Watermark dialog box.

---

**Syntax** WatermarkDlg ()

---

**Route** Layout, Watermark.

---

## WatermarkPlacementDlg

---

**Purpose** WatermarkPlacementDlg displays the Placement dialog box.

---

**Syntax** WatermarkPlacementDlg ()

---

**Route** Layout, Watermark, Create, Placement.



---

## WidowOrphan

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| <b>Purpose</b>    | WidowOrphan specifies whether Widow/Orphan is on.                                               |
| <b>Syntax</b>     | WidowOrphan (State: <i>Numeric Equivalent</i> or <i>Enumerated Type</i> )                       |
| <b>Parameters</b> |                                                                                                 |
| <i>State</i>      | Numeric Equivalent or Enumerated Type: Specifies whether Widow/Orphan is on.<br>0 Off!<br>1 On! |
| <b>Route</b>      | Layout, Page, Keep Text Together, select option in Widow/Orphan box.                            |
| <b>Example</b>    | To turn on Widow/Orphan, the command is:<br>WidowOrphan (State: On!)                            |
| <b>See Also</b>   | • ?WidowOrphan                                                                                  |

---

## WindowCascade

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| <b>Purpose</b> | WindowCascade overlaps document windows so the title bar of each window displays. |
| <b>Syntax</b>  | WindowCascade ()                                                                  |
| <b>Route</b>   | Window, Cascade.                                                                  |

---

## WindowNext

|                |                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | WindowNext moves the insertion point to the next open dialog box or document window.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | WindowNext ()                                                                                                                                                                                |

---

## WindowPrevious

|                |                                                                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | WindowPrevious moves the insertion point to the previous open dialog box or document window.<br><br>IMPORTANT: This command is not recordable. To use this command, you must type it into the macro. |
| <b>Syntax</b>  | WindowPrevious ()                                                                                                                                                                                    |

---

## WindowTile

|                |                                                                                              |
|----------------|----------------------------------------------------------------------------------------------|
| <b>Purpose</b> | WindowTile reduces open document windows and displays them side by side with no overlapping. |
| <b>Syntax</b>  | WindowTile ()                                                                                |
| <b>Route</b>   | Window, Tile.                                                                                |

---

## WordCountDlg

|                |                                                            |
|----------------|------------------------------------------------------------|
| <b>Purpose</b> | WordCountDlg displays the Document Information dialog box. |
| <b>Syntax</b>  | WordCountDlg ()                                            |
| <b>Route</b>   | File, Document Info.                                       |

---

## WordLetterspaceDlg

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| <b>Purpose</b> | WordLetterspaceDlg displays the Word Spacing and Letterspacing dialog box. |
| <b>Syntax</b>  | WordLetterspaceDlg ()                                                      |
| <b>Route</b>   | Layout, Typesetting, Word/Letterspacing.                                   |

---

# WordLetterSpacing

---

**Purpose** WordLetterSpacing specifies the display spacing between words and letters. IMPORTANT: Normal is the specification suggested by the font manufacturer; Optimal is the specification suggested by WordPerfect Corporation.

---

**Syntax** WordLetterSpacing (CharacterWidth: *Numeric Equivalent* or *Enumerated Type*, SpaceWidth: *Numeric Equivalent* or *Enumerated Type*)

---

## Parameters

*CharacterWidth* Numeric Equivalent or Enumerated Type: Specifies spacing between words. This parameter also accepts a numeric expression to specify a percentage of the WordPerfect default Optimal.

0 Normal!  
100 Optimal!

*SpaceWidth* Numeric Equivalent or Enumerated Type: Specifies spacing between letters. This parameter also accepts a numeric expression to specify a percentage of the WordPerfect default Optimal.

0 Normal!  
100 Optimal!

---

**Route** Layout, Typesetting, Word/Letterspacing, specify options.

---

**Example** To specify normal word spacing with letter spacing 80 percent of Optimal, the command is:

WordLetterSpacing (CharacterWidth: Normal!; SpaceWidth: 80)

---

**See Also** ♦ ?CharSpaceWidth

---

---

# WordSpacingJustificationLimits

---

**Purpose** WordSpacingJustificationLimits applies when Full justification is on, adjusting spaces between words to justify text on a line. When a word spacing limit is reached, character spacing begins.

---

**Syntax** WordSpacingJustificationLimits (Compression: *Numeric Expression*; Expansion: *Numeric Expression*)

---

---

**Parameters**

*Compression*

Numeric Expression: Specifies a minimum percentage.

*Expansion*

Numeric Expression: Specifies a maximum percentage.

---

**Route**

Layout, Typesetting, Word/Letterspacing, specify settings in the Compressed To and Expanded To text boxes.

---

**Example**

To specify minimum and maximum justification limits of 40% and 300% respectively, the command is:

`WordSpacingJustificationLimits (Compression: 40; Expansion: 300)`

---

**See Also**

♦ `?SpaceExpansion`

---

---

## WPCharactersDlg

**Purpose**

WPCharactersDlg displays the WordPerfect Characters dialog box.

---

**Syntax**

WPCharactersDlg ()

---

**Route**

Insert, Character.

---

---

## WPDraw

**Purpose**

WPDraw opens Draw.

---

**Syntax**

WPDraw ()

---

**Route**

Graphics, Draw.

---

**See Also**

♦ `GraphicCreateChart`

---

---

## WPGrammatik

**Purpose**

WPGrammatik opens Grammatik.

---

---

|                 |                           |
|-----------------|---------------------------|
| <b>Syntax</b>   | WPGrammatik ()            |
| <b>Route</b>    | Tools, Grammatik.         |
| <b>See Also</b> | ♦ WPSpeller ♦ WPThesaurus |

---

## WPSpeller

---

|                 |                                            |
|-----------------|--------------------------------------------|
| <b>Purpose</b>  | WPSpeller displays the Speller dialog box. |
| <b>Syntax</b>   | WPSpeller ()                               |
| <b>Route</b>    | Tools, Speller.                            |
| <b>See Also</b> | ♦ WPGrammatik ♦ WPThesaurus                |

---

## WPThesaurus

---

|                 |                                                |
|-----------------|------------------------------------------------|
| <b>Purpose</b>  | WPThesaurus displays the Thesaurus dialog box. |
| <b>Syntax</b>   | WPThesaurus ()                                 |
| <b>Route</b>    | Tools, Thesaurus.                              |
| <b>See Also</b> | ♦ WPGrammatik ♦ WPSpeller                      |

---

## WritingToolsDisable

---

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| <b>Purpose</b>  | WritingToolsDisable specifies whether to disable Writing Tools.          |
| <b>Syntax</b>   | WritingToolsDisable ()                                                   |
| <b>Route</b>    | File, Preferences, Writing Tools icon, select or deselect writing tools. |
| <b>See Also</b> | ♦ ?WritingToolsDisabledState                                             |

---

---

## ZoomDlg

|                 |                                       |
|-----------------|---------------------------------------|
| <b>Purpose</b>  | ZoomDlg displays the Zoom dialog box. |
| <b>Syntax</b>   | ZoomDlg ()                            |
| <b>Route</b>    | View, Zoom.                           |
| <b>See Also</b> | ♦ ZoomToPageWidth ♦ ?Zoom             |

---

## ZoomToFullPage

|                |                                                    |
|----------------|----------------------------------------------------|
| <b>Purpose</b> | ZoomToFullPage displays a document as a full page. |
| <b>Syntax</b>  | ZoomToFullPage ()                                  |
| <b>Route</b>   | View, Zoom, select Full Page.                      |

---

## ZoomToMarginWidth

|                |                                                                |
|----------------|----------------------------------------------------------------|
| <b>Purpose</b> | ZoomToMarginWidth displays a document between margin settings. |
| <b>Syntax</b>  | ZoomToMarginWidth ()                                           |
| <b>Route</b>   | View, Zoom, select Margin Width.                               |

---

## ZoomToPageWidth

|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| <b>Purpose</b>  | ZoomToPageWidth displays the current document between the left and right margins. |
| <b>Syntax</b>   | ZoomToPageWidth ()                                                                |
| <b>Route</b>    | View, Zoom, select Page Width.                                                    |
| <b>See Also</b> | ♦ ZoomDlg                                                                         |

# System Variables Reference

.....

## System Variables Reference



---

## ?Align

---

**Purpose** ?Align contains a value representing the current character justification.

| Value | Description        |
|-------|--------------------|
| 0     | Left, Full, or All |
| 64    | Center             |
| 128   | Right              |

---

## ?AutoUpdate

---

**Purpose** ?AutoUpdate contains a value representing whether Update on Retrieve is on or off for spreadsheet/database links.

| Value | Description |
|-------|-------------|
| 0     | Off         |
| 1     | On          |

---

## ?BaselineMode

---

**Purpose** ?BaselineMode contains a value representing whether Baseline Placement for Typesetting is on or off.

| Value | Description               |
|-------|---------------------------|
| 0     | Baseline Placement is off |
| 1     | Baseline Placement is on  |

**See Also** ♦ BaselinePlacement

---

---

## ?BetweenTableCodes

---

**Purpose** ?BetweenTableCodes contains a value representing whether the insertion point is in a table.

| Value | Description                       |
|-------|-----------------------------------|
| False | Insertion point is not in a table |
| True  | Insertion point is in a table     |

---

## ?Binding

---

### Purpose

?Binding contains a value representing the binding edge for the current page. The value of this variable is one of the numeric equivalents or enumerated types listed for the Edge parameter of Binding.

For example, the following macro causes the computer to beep if the binding edge is set as the left edge:

```
IF (?Binding=0)
 BEEP
ENDIF
```

---

### See Also

- ♦ Binding

---

## ?BlockActive

---

### Purpose

?BlockActive contains a value representing whether block is on.

| Value | Description    |
|-------|----------------|
| 0     | Block not on   |
| 1     | CharMode       |
| 2     | WordMode       |
| 4     | SentenceMode   |
| 8     | ParagraphMode  |
| 32    | RectangleMode  |
| 64    | TabularColumn  |
| 128   | DocMode        |
| 256   | CharNoCellMode |

---

## ?BlockProtect

---

### Purpose

?BlockProtect contains a value representing whether Block Protect is on or off.

| Value | Description          |
|-------|----------------------|
| False | Block Protect is off |
| True  | Block Protect is on  |

---

### See Also

- ♦ BlockProtect

---

## ?BookmarkList

---

**Purpose**                   ?BookmarkList contains the name of the first bookmark in a document.

---

## ?Border

---

**Purpose**                   ?Border contains a value representing the current border style. The value of this variable is one of the numeric equivalents or enumerated types listed for the Style parameter of BorderStyleEdit. This variable must be preceded by a command which specifies a border or border style, such as TextBorderEdit, and followed by a closing command such as TextBorderEnd.

For example, the following macro causes the computer to beep if the current border style is Dashed Border:

```
TextBorderEdit (BorderType: Paragraph!)
IF (?Border=3)
BEEP
ENDIF
TextBorderEnd (State: Save!)
```

---

**See Also**                   ♦ BorderStyleEdit

---

---

## ?BorderBottomLine

---

**Purpose**                   ?BorderBottomLine contains a value representing the bottom line style of a border. The value of this variable is either one of the numeric equivalents or enumerated types listed for BorderBottomLine, or the name of a custom-made line style. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the bottom line style specified for a paragraph border:

```
TextBorderEdit (BorderType: ParagraphBorder!)
Type (?BorderBottomLine)
TextBorderEnd (State: Save!)
```

---

**See Also**                   ♦ BorderBottomLine

---

---

## ?BorderColorBlue

---

### Purpose

?BorderColorBlue contains a value representing the amount of blue in a border color. Values range from 0-255, with 0 indicating no blue and 255 indicating solid blue. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the amount of blue specified for a paragraph border:

```
TextBorderEdit (Border Type: Paragraph!)
Type (?BorderColorBlue)
TextBorderEnd (State: Save!)
```

---

### See Also

- BorderColor

---

## ?BorderColorGreen

---

### Purpose

?BorderColorGreen contains a value representing the amount of green in a border color. Values range from 0-255, with 0 indicating no green and 255 indicating solid green. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the amount of green specified for a paragraph border:

```
TextBorderEdit (Border Type: Paragraph!)
Type (?BorderColorGreen)
TextBorderEnd (State: Save!)
```

---

### See Also

- BorderColor

---

## ?BorderColorRed

---

### Purpose

?BorderColorRed contains a value representing the amount of red in a border color. Values range from 0-255, with 0 indicating no red and 255 indicating solid red. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the amount of red specified for a paragraph border:

```
TextBorderEdit (Border Type: Paragraph!)
Type (?BorderColorRed)
TextBorderEnd (State: Save!)
```

---

**See Also**

- ◆ `BorderColor`

---

## ?BorderColorShading

**Purpose**

?BorderColorShading contains a numeric expression representing the shading percentage of a border color. This variable must be preceded by either `TextBorderCreate` or `TextBorderEdit` and followed by `TextBorderEnd`.

For example, the following macro types a value representing the shading of a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderColorShading)
TextBorderEnd (State: Save!)
```

---

**See Also**

- ◆ `BorderColor`

---

## ?BorderCornerRadius

**Purpose**

?BorderCornerRadius contains a measurement expression representing the corner radius of a border. This variable must be preceded by either `TextBorderCreate` or `TextBorderEdit` and followed by `TextBorderEnd`.

For example, the following macro types a value representing the corner radius specified for a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderCornerRadius)
TextBorderEnd (State: Save!)
```

---

**See Also**

- ◆ `BorderCornerRadius`

---

## ?BorderInsideSpacingBottom

**Purpose**

?BorderInsideSpacingBottom contains a measurement expression representing the bottom spacing inside a border. This variable must be preceded by either `TextBorderCreate` or `TextBorderEdit` and followed by `TextBorderEnd`.

For example, the following macro types the measurement of the bottom inside spacing specified for a paragraph border:

TextBorderEdit (BorderType: Paragraph!)  
Type (?BorderInsideSpacingBottom)  
TextBorderEnd (State: Save!)

---

**See Also**

- ◆ BorderInsideSpacing

---

## ?BorderInsideSpacingTop

**Purpose**

?BorderInsideSpacingTop contains a measurement expression representing the top spacing inside a border. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a measurement of the top inside spacing specified for a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderInsideSpacingTop)
TextBorderEnd (State: Save!)
```

---

**See Also**

- ◆ BorderInsideSpacing

---

## ?BorderLeftLine

**Purpose**

?BorderLeftLine contains a value representing the left line style of a border. The value of this variable is either one of the numeric equivalents or enumerated types listed for BorderLeftLine, or the name of a custom-made line style. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the left line style specified for a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderLeftLine)
TextBorderEnd (State: Save!)
```

---

**See Also**

- ◆ BorderLeftLine

---

## ?BorderRightLine

---

### Purpose

?BorderRightLine contains a value representing the right line style of a border. The value of this variable is either one of the numeric equivalents or enumerated types listed for BorderRightLine, or the name of a custom-made line style. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the right line style specified for a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderRightLine)
TextBorderEnd (State: Save!)
```

---

### See Also

- ◆ BorderRightLine

---

## ?BorderSeparatorLine

---

### Purpose

?BorderSeparatorLine contains a value representing the separator line style of a border. The value of this variable is either one of the numeric equivalents or enumerated types listed for BorderSeparatorLine, or the name of a custom-made line style. This variable must be preceded by either TextBorderCreate or TextBorderEdit and followed by TextBorderEnd.

For example, the following macro types a value representing the separator line style specified for a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderSeparatorLine)
TextBorderEnd (State: Save!)
```

---

### See Also

- ◆ BorderSeparatorLine

---

## ?BorderSetSpacing

---

### Purpose

?BorderSetSpacing contains a value representing whether Auto Spacing is on or off for the current border.

| Value | Description         |
|-------|---------------------|
| 0     | Auto Spacing is off |
| 1     | Auto Spacing is on  |

---

**See Also**

- ◆ `BorderSetSpacing`

---

## ?BorderTopLine

**Purpose**

?BorderTopLine contains a value representing the top line style of a border. The value of this variable is either one of the numeric equivalents or enumerated types listed for `BorderTopLine`, or the name of a custom-made line style. This variable must be preceded by either `TextBorderCreate` or `TextBorderEdit` and followed by `TextBorderEnd`.

For example, the following macro types a value representing the top line style specified for a paragraph border:

```
TextBorderEdit (BorderType: Paragraph!)
Type (?BorderTopLine)
TextBorderEnd (State: Save!)
```

---

**See Also**

- ◆ `BorderTopLine`

---

## ?BorderUseBorderColor

**Purpose**

?BorderUseBorderColor contains a value representing whether the border color option is used for a border. This variable must be preceded by either `TextBorderCreate` or `TextBorderEdit` and followed by `TextBorderEnd`.

| Value | Description      |
|-------|------------------|
| 0     | Do not use color |
| 1     | Use color        |

---

**See Also**

- ◆ `BorderUseBorderColor`



---

## ?BoxAttachTo

---

### Purpose

?BoxAttachTo contains a value representing whether a graphics box is attached to a page, a paragraph, or a character. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxAttachTo. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if box 1 is attached to a page:

```
BoxEdit (BoxNumber: 1)
IF (?BoxAttachTo=2)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

### See Also

- ♦ BoxAttachTo

---

## ?BoxCaptionDefaultInitialStyle

---

### Purpose

?BoxCaptionDefaultInitialStyle contains a value representing the current box caption initial style. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxCaptionInitialStyle. This variable must be preceded by a command which identifies a box or box style, such as BoxEdit, and followed by a closing command such as BoxEnd.

For example, the following macro types a value representing the initial caption style specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxCaptionDefaultInitialStyle)
BoxEnd (State: Save!)
```

---

### See Also

- ♦ BoxCaptionInitialStyle

---

## ?BoxCaptionDefaultNumberStyle

---

### Purpose

?BoxCaptionDefaultNumberStyle contains a value representing the current box caption number style. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxCaptionNumberStyle. This variable must be preceded by a command which identifies a box or box style, such as BoxEdit, and followed by a closing command such as BoxEnd.

For example, the following macro types a value representing the caption number style specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxCaptionDefaultNumberStyle)
BoxEnd (State: Save!)
```

---

**See Also**

- ◆ BoxCaptionNumberStyle

---

## ?BoxCaptionFormatting

**Purpose**

?BoxCaptionFormatting contains a value representing the width of a graphics box caption. If the width is set to Auto, the value of this variable is 2. Otherwise, the value is either a measurement expression in the current unit of measure or a percentage of the box width or height. This command must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types a value representing the caption width specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxCaptionFormatting)
BoxEnd (State: Save!)
```

---

## ?BoxCaptionPositionAlignment

**Purpose**

?BoxCaptionPositionAlignment contains a value representing the alignment of a graphics box caption. The value of this variable is one of the numeric equivalents or enumerated types listed for the Alignment parameter of BoxCaptionPosition. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if top alignment is specified for the caption of box 1 :

```
BoxEdit (BoxNumber:1)
IF (?BoxCaptionPositionAlignment=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also**

- ◆ BoxCaptionPosition

---

## ?BoxCaptionPositionBorder

---

### Purpose

?BoxCaptionPositionBorder contains a value representing the position of a graphics box caption relative to the border. The value of this variable is one of the numeric equivalents or enumerated types listed for the Border parameter of BoxCaptionPosition. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if the caption for box 1 is inside the border:

```
BoxEdit (BoxNumber: 1)
IF (?BoxCaptionPositionBorder=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

### See Also

- ♦ BoxCaptionPosition

---

## ?BoxCaptionPositionOffset

---

### Purpose

?BoxCaptionPositionOffset contains a value representing the offset position of a graphics box caption. The value of this variable is either a measurement expression in the current unit of measure or a percentage of the box width or height. This variable must be preceded by BoxCreate or BoxEdit and followed by BoxEnd.

For Example:

```
BoxEdit (BoxNum: 1)
Type (?BoxCaptionPositionOffset)
BoxEnd (State: Save!)
```

---

## ?BoxCaptionPositionSide

---

### Purpose

?BoxCaptionPositionSide contains a value representing the side of a box where a caption is located. This command must be preceded by BoxCreate or BoxEdit and followed by BoxEnd, or be played from a box edit screen with a box selected. The value of this variable is one of the numeric equivalents or enumerated types listed for the Side parameter of BoxCaptionPosition.

For Example:

```
BoxEdit (BoxNum:1)
IF (?BoxCaptionPosition=4)
BEEP
ENDIF
```

---

## ?BoxCaptionRotation

### Purpose

?BoxCaptionRotation contains a value representing the rotation of a box caption. This command must be preceded by BoxCreate or BoxEdit and followed by BoxEnd, or be played from a box edit screen with a box selected. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxCaptionRotation.

For Example:

```
BoxEdit (BoxNum:1)
IF (?BoxCaptionRotation=0)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

## ?BoxChangeLineHeight

### Purpose

?BoxChangeLineHeight specifies whether a graphics box attached to a character position automatically changes the text line height. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description |
|-------|-------------|
| 0     | No change   |
| 1     | Change      |

### See Also

- ♦ BoxChangeLineHeight

---

## ?BoxContentFilename

### Purpose

?BoxContentFilename contains the name of a text file or image file contained in a graphics box. The value of this variable is a filename. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types the name of the file contained in box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxContentFilename)
BoxEnd (State: Save!)
```

---

## ?BoxContentHorizontalPosition

---

**Purpose** ?BoxContentHorizontalPosition contains a value representing the horizontal position of graphics box contents. The value of this variable is one of the numeric equivalents or enumerated types listed for the Horizontal parameter of BoxContentPosition. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if the horizontal position specified for the contents of box 1 is left:

```
BoxEdit (BoxNumber: 1)
IF (?BoxContentHorizontalPosition=0)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxContentPosition

---

---

## ?BoxContentInternal

---

**Purpose** ?BoxContentInternal contains a value representing whether an image is stored in the current document or on a disk.

| Value | Description |
|-------|-------------|
| 0     | In document |
| 1     | On disk     |

---

## ?BoxContentPreserveAspectRatio

---

**Purpose** ?BoxContentPreserveAspectRatio contains a value representing whether the image width/height ratio is preserved when the position of a box changes. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description   |
|-------|---------------|
| 0     | Not preserved |
| 1     | Preserved     |

---

**See Also** ♦ BoxContentPreserveAspectRatio

---

---

## ?BoxContentType

---

### Purpose

?BoxContentType contains a value representing the type of content in a graphics box. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxContentType. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if box 1 contains text:

```
BoxEdit (BoxNumber: 1)
IF (?BoxContentType=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

### See Also

- ♦ BoxContentType

---

## ?BoxContentVerticalPosition

---

### Purpose

?BoxContentVerticalPosition contains a value representing the vertical position of graphics box content. The value of this variable is one of the numeric equivalents or enumerated types listed for the Vertical parameter of BoxContentPosition. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if the vertical position specified for the contents of box 1 is specified as bottom:

```
BoxEdit (BoxNumber: 1)
IF (?BoxContentVerticalPosition=4)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

### See Also

- ♦ BoxContentPosition

---

## ?BoxCount

---

### Purpose

?BoxCount contains the number of boxes in the current document.

---

## ?BoxCounter

---

**Purpose**                   ?BoxCounter contains a value representing the counter type of a graphics box or graphics box style. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxCounter. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd, or be preceded by either BoxStyleCreate or BoxStyleEdit and followed by BoxStyleEnd.

For example, the following macro causes the computer to beep if the counter for box 1 is a figure counter:

```
BoxEdit (BoxNumber: 1)
IF (?BoxCounter=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also**                   ♦ BoxCounter

---

## ?BoxEquationColorBlue

---

**Purpose**                   ?BoxEquationColorBlue contains a value representing whether an equation color contains blue. Values range from 0 to 255, with 0 indicating no blue and 255 indicating solid blue. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd, or be preceded by either BoxStyleCreate or BoxStyleEdit and followed by BoxStyleEnd.

For example, the following macro types a value representing the amount of blue in the equation color specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxEquationColorBlue)
BoxEnd (State: Save!)
```

---

**See Also**                   ♦ BoxEquationColor

---

## ?BoxEquationColorGreen

---

**Purpose**                   ?BoxEquationColorGreen contains a value representing whether an equation color contains green. Values range from 0 to 255, with 0 indicating no green and 255 indicating solid green. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd, or be preceded by either BoxStyleCreate or BoxStyleEdit and followed by BoxStyleEnd.

For example, the following macro types a value representing the amount of green in the equation color specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxEquationColorGreen)
BoxEnd (State: Save!)
```

---

**See Also**

- ♦ `BoxEquationColor`

---

## ?BoxEquationColorRed

**Purpose**

?BoxEquationColorRed contains a value representing whether an equation color contains red. Values range from 0 to 255, with 0 indicating no red and 255 indicating solid red. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`, or be preceded by either `BoxStyleCreate` or `BoxStyleEdit` and followed by `BoxStyleEnd`.

For example, the following macro types a value representing the amount of red in the equation color specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxEquationColorRed)
BoxEnd (State: Save!)
```

---

**See Also**

- ♦ `BoxEquationColor`

---

## ?BoxEquationColorShading

**Purpose**

?BoxEquationColorShading contains a value representing the shade of an equation color. The value of this variable is a percentage, with 100 indicating the darkest shade of a color. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`, or be preceded by `BoxStyleCreate` or `BoxStyleEdit` and followed by `BoxStyleEnd`.

For example, the following macro types a value representing the shading of the equation color specified for box 1:

```
BoxEdit (BoxNumber: 1)
Type (?BoxEquationColorShading)
BoxEnd (State: Save!)
```

---

**See Also**

- ♦ `BoxEquationColor`



---

## ?BoxEquationFont

---

**Purpose** ?BoxEquationFont contains a specified font for an equation box. The value of this variable is the name of a font. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd, or be preceded by BoxStyleCreate or BoxStyleEdit and followed by BoxStyleEnd.

For example, the following macro types the name of the equation font specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxEquationFont
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also** ♦ BoxEquationDefaultFont

---

---

## ?BoxEquationFontSize

---

**Purpose** ?BoxEquationFontSize contains a value representing an equation box font size. The value of this variable is a measurement expression in the current unit of measure. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd, or be preceded by either BoxStyleCreate or BoxStyleEdit and followed by BoxStyleEnd.

For example, the following macro types a value representing the equation font size specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxEquationFontSize
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also** ♦ BoxEquationFontSize

---

---

## ?BoxHeight

---

**Purpose** ?BoxHeight contains a value representing the height of a graphics box. If the height is set to Auto, the value of this variable is 2. Otherwise, the value is a measurement expression in WordPerfect units (1200ths of an inch). This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types the height of box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxHeight
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also**

- ♦ BoxHeight

---

## ?BoxHorizontalAlignment

**Purpose**

?BoxHorizontalAlignment contains a value representing the horizontal position of a graphics box attached to a page or a paragraph. The value of this variable is one of the numeric equivalents or enumerated types listed for the Position parameter of BoxHorizontalAlignment. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if full horizontal alignment is specified for box 1:

```
BoxEdit (BoxNumber: 1)
IF (?BoxHorizontalAlignment=7)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also**

- ♦ BoxHorizontalAlignment

---

## ?BoxHorizontalAlignTo

**Purpose**

?BoxHorizontalAlignTo contains a value representing whether a graphics box attached to a page is aligned to the margins or to a column or columns. The value of this variable is one of the numeric equivalents or enumerated types listed for the Alignment parameter of BoxHorizontalAlignment. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if box 1 is aligned to the margins:

```
BoxEdit (BoxNumber: 1)
IF (?BoxHorizontalAlignTo=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also**

- ◆ `BoxHorizontalAlignment`

---

## ?BoxHorizontalOffset

**Purpose**

?BoxHorizontalOffset contains a value representing the right or left position of a graphics box relative to an alignment point. The value of this variable is a measurement expression in WordPerfect units (1200ths of an inch). This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types the offset measurement specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxHorizontalOffset
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also**

- ◆ `BoxHorizontalAlignment`

---

## ?BoxImageBlackWhiteThreshold

**Purpose**

?BoxImageBlackWhiteThreshold contains a value representing a threshold where colors and/or grays convert to black or white. Colors or grays below the threshold are converted to black; those above are converted to white. Values range from 1 to 255, with 127 as the default. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types a value representing the black/white threshold specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxImageBlackWhiteThreshold
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also**

- ◆ `BoxImageBlackWhiteThreshold`

---

## ?BoxImageBrightness

---

### Purpose

?BoxImageBrightness contains a value representing the brightness of an image color. Values range from -1.0 (all black) to 1.0 (all white). This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types a value representing the brightness specified for the image contained in box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxImageBrightness
BoxEnd (State: Save!)
Type (Var)
```

---

### See Also

- ♦ BoxImageBrightness

---

## ?BoxImageContrast

---

### Purpose

?BoxImageContrast contains a value representing image contrast. Values range from -1.0 (slight contrast) to 1.0 (substantial contrast). This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types a value representing the contrast specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxImageContrast
BoxEnd (State: Save!)
Type (Var)
```

---

### See Also

- ♦ BoxImageContrast

---

## ?BoxImageDitherMethod

---

### Purpose

?BoxImageDitherMethod contains a value representing a dither method. The value of this variable is one of the numeric equivalents or enumerated types listed for the Method parameter of BoxImageDitherMethod. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if the dither method specified for box 1 is halftoning:

```
BoxEdit (BoxNumber: 1)
IF (?BoxImageDitherMethod=1)
BEEP
```

```
ENDIF
BoxEnd (State: Save!)
```

---

## ?BoxImageDitherSource

**Purpose** ?BoxImageDitherSource contains a value representing a dither source. The value of this variable is one of the numeric equivalents or enumerated types listed for the Source parameter of BoxImageDitherMethod. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For Example:

```
BoxEdit (BoxNumber: 1)
IF (?BoxImageDitherSource=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

## ?BoxImageFill

**Purpose** ?BoxImageFill contains a value representing an image fill type. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxImageFill. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if a transparent fill type is specified for box 1:

```
BoxEdit (BoxNumber: 1)
IF (?BoxImageFill=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also** ♦ BoxImageFill

---

## ?BoxImageFlipX

**Purpose** ?BoxImageFlipX contains a value representing whether the horizontal axis of an image is reversed. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description  |
|-------|--------------|
| 0     | Not reversed |
| 1     | Reversed     |

---

**See Also** ♦ BoxImageFlipX

---

## ?BoxImageFlipY

**Purpose** ?BoxImageFlipY contains a value representing whether the vertical axis of an image is reversed. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description  |
|-------|--------------|
| 0     | Not reversed |
| 1     | Reversed     |

---

**See Also** ♦ BoxImageFlipY

---

## ?BoxImageHalftoneAngle0

**Purpose** ?BoxImageHalftoneAngle0 contains the image halftone angle. The default is generally 45, but is dependent upon the selected printer.

---

## ?BoxImageHalftoneAngle1

**Purpose** ?BoxImageHalftoneAngle1 contains the image halftone angle 1.

---

## ?BoxImageHalftoneAngle2

**Purpose** ?BoxImageHalftoneAngle2 contains the image halftone angle 2.

---

## ?BoxImageHalftoneAngle3

---

**Purpose**                   ?BoxImageHalftoneAngle3 contains the image halftone angle 3.

---

---

## ?BoxImageHalftoneScreenLPI

---

**Purpose**                   ?BoxImageHalftoneScreenLPI contains the halftone lines per inch. The default is 60.

---

---

## ?BoxImageInvertColors

---

**Purpose**                   ?BoxImageInvertColors contains a value representing whether image colors have been inverted to their respective complementary colors. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description  |
|-------|--------------|
| 0     | Not inverted |
| 1     | Inverted     |

---

**See Also**               • BoxImageInvertColors

---

---

## ?BoxImageMonochrome

---

**Purpose**                   ?BoxImageMonochrome contains a value representing whether a color image converts to black and white. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description    |
|-------|----------------|
| 0     | Not monochrome |
| 1     | Monochrome     |

---

**See Also**               • BoxImageMonochrome

---

---

## ?BoxImagePageBackground

---

**Purpose** ?BoxImagePageBackground contains a value representing whether any background colors or gradients saved with an image are visible.

| Value | Description            |
|-------|------------------------|
| 0     | Background not visible |
| 1     | Background visible     |

**See Also** ♦ BoxImageShowBackground

---

---

## ?BoxImageRotation

---

**Purpose** ?BoxImageRotation contains a value representing a rotation degree. Values range from 0.0 to 359.0. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types the rotation degree specified for the image contained in box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxImageRotation
BoxEnd (State: Save!)
Type (Var)
```

**See Also** ♦ BoxImageRotate

---

---

## ?BoxImageScalingX

---

**Purpose** ?BoxImageScalingX contains a value representing a scaling width. The value of this variable ranges from 0.01 to 9.99. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types a value representing the scaling width specified for the image contained in box 1:

```
BoxEdit (BoxNumber: 1)
IF (?BoxImageScalingX=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```



---

**See Also** ♦ `BoxImageScaling`

---

## ?BoxImageScalingY

**Purpose** `?BoxImageScalingY` contains a value representing a scaling height. The value of this variable ranges from 0.01 to 9.99. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types a value representing the scaling height specified for the image contained in box 1:

```
BoxEdit (BoxNumber: 1)
IF (?BoxImageScaling=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also** ♦ `BoxImageScaling`

---

## ?BoxImageTranslationX

**Purpose** `?BoxImageTranslationX` contains a value representing the horizontal position of an image within a box. Values range from -5.0 to 5.0, with the default at 0.0. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types a value representing the horizontal position of the contents of box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxImageTranslationX
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also** ♦ `BoxImageMove`

---

## ?BoxImageTranslationY

**Purpose** `?BoxImageTranslationY` contains a value representing the vertical position of an image within a box. Values range from -5.0 to 5.0, with the default at 0.0. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types a value representing the vertical position of the contents of box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxImageTranslationY
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also**

- ♦ `BoxImageMove`

---

## ?BoxLeftColumn

**Purpose**

?BoxLeftColumn contains a value representing the left column to which a graphics box aligns. The value of this variable is the number of a column. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types the number of the left column to box 1 is aligned to:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxLeftColumn
BoxEnd (State: Save!)
Type (Var)
```

---

## ?BoxNumber

**Purpose**

?BoxNumber contains the current graphics box number. The value of this variable is the number of the box following the insertion point.

---

## ?BoxOverlap

**Purpose**

?BoxOverlap contains a value representing whether a graphics box may overlap other boxes. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

| Value | Description |
|-------|-------------|
| 0     | No overlap  |
| 1     | Overlap     |

---

**See Also**

- ♦ `BoxOverlap`

---

## ?BoxPageOffset

**Purpose**

?BoxPageOffset contains a value representing whether a box stays on the page to which it was attached or moves with text. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

| Value | Description           |
|-------|-----------------------|
| 0     | Does not stay on page |
| 1     | Stays on page         |

---

**See Also**

- ♦ `BoxPageOffset`

---

## ?BoxRightColumn

**Purpose**

?BoxRightColumn contains a value representing the right column to which a graphics box aligns. The value of this variable is the number of a column. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`.

For example, the following macro types the number of the right column box 1 is aligned to:

```
BoxEdit (BoxNumber: 1)
Var=?BoxRightColumn
BoxEnd (State: Save!)
Type (Var)
```

---

## ?BoxStyleName

**Purpose**

?BoxStyleName contains the name of a box style. The value of this variable is either one of the numeric equivalents or enumerated types listed for `BoxStyleEdit`, or the name of a user-created style. This variable must be preceded by either `BoxCreate` or `BoxEdit` and followed by `BoxEnd`, or be preceded by either `BoxStyleCreate` or `BoxStyleEdit` and followed by `BoxStyleEnd`.

For example, the following macro types the name of the box style specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var=?BoxStyleName
BoxEnd (State: Save!)
Type (Var)
```

---

## ?BoxTextAngle

### Purpose

?BoxTextAngle contains a value representing the degree of rotation for text in a box. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxTextAngle. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if the text in box 1 is rotated 90 degrees:

```
BoxEdit (BoxNumber: 1)
IF (?BoxTextAngle=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

### See Also

- ♦ BoxTextAngle

---

## ?BoxTextDefaultStyle

### Purpose

?BoxTextDefaultStyle contains the default style of box text. The value of this variable is one of the numeric equivalents or enumerated types listed for BoxTextDefaultStyle. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if the default text style specified for box 1 is Doc Style.

```
BoxEdit (BoxNumber: 1)
IF (?BoxTextDefaultStyle=33)
BEEP
ENDIF
BoxEnd (State: Save!)
```

### See Also

- ♦ BoxTextDefaultStyle

---

## ?BoxVerticalAlignment

### Purpose

?BoxVerticalAlignment contains a value representing the vertical position of a graphics box attached to a page or character. The value of this variable is one of the numeric equivalents or enumerated types listed for the Alignment parameter of BoxVerticalAlignment. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For Example:

```
BoxEdit (BoxNumber: 1)
IF (?BoxVerticalAlignment=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also**

- ◆ BoxVerticalAlignment

---

## ?BoxVerticalAlignTo

---

**Purpose**

?BoxVerticalAlignTo contains a value representing whether a box is aligned to margins.

| Value | Description        |
|-------|--------------------|
| 1     | Aligned to margins |
| 2     | Fixed position     |

---

## ?BoxVerticalOffset

---

**Purpose**

?BoxVerticalOffset contains a value representing the vertical position of a graphics box relative to an alignment point. The value of this variable is a measurement expression in WordPerfect units (1200th of an inch). This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types a value representing the vertical offset specified for box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxVerticalOffset
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also**

- ◆ BoxVerticalAlignment

---

## ?BoxWidth

---

**Purpose**

?BoxWidth contains a value representing the width of a graphics box. If the width is set to Auto, the value of this variable is one of the numeric equivalents or enumerated types listed for BoxWidth. Otherwise, the value is a measurement expression in WordPerfect units (1200ths of an inch). This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example:, the following macro types a value representing the width of box 1:

```
BoxEdit (BoxNumber: 1)
Var:=?BoxWidth
BoxEnd (State: Save!)
Type (Var)
```

---

**See Also**

- ♦ BoxWidth

---

## ?BoxWrapping

**Purpose**

?BoxWrapping contains a value representing the direction of text flow around a graphics box. The value of this variable is one of the numeric equivalents or enumerated types of BoxTextFlow. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro causes the computer to beep if text flows to the left of box 1:

```
BoxEdit (BoxNumber: 1)
IF (?BoxWrapping=1)
BEEP
ENDIF
BoxEnd (State: Save!)
```

---

**See Also**

- ♦ BoxTextFlow

---

## ?BoxWrappingType

**Purpose**

?BoxWrappingType contains a value representing whether the text outside a graphics box flows around an image. This variable must be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description           |
|-------|-----------------------|
| 0     | Text is not contoured |
| 1     | Text is contoured     |

---

## ?ButtonBarActive

**Purpose** ?ButtonBarActive contains a value representing whether the Button Bar is active in the current document.

| Value | Description |
|-------|-------------|
| False | Not active  |
| True  | Active      |

---

## ?Cell

**Purpose** ?Cell contains a value representing the current table cell. For example, if the insertion point is in the first cell of a table, this variable contains the value A1.

---

## ?CellAlignment

**Purpose** ?CellAlignment contains a value representing the vertical alignment of the current cell.

| Value | Description |
|-------|-------------|
| 1     | Top         |
| 2     | Center      |
| 4     | Bottom      |

---

## ?CellBold

**Purpose** ?CellBold indicates whether the cell font attribute is Bold.

| Value | Description |
|-------|-------------|
| False | Not Bold    |
| True  | Bold        |

---

## ?CellDoubleUnderline

---

|                |                                                                                     |
|----------------|-------------------------------------------------------------------------------------|
| <b>Purpose</b> | ?CellDoubleUnderline indicates whether the cell font attribute is Double Underline. |
| <b>Value</b>   | <b>Description</b>                                                                  |
| False          | Not Double Underline                                                                |
| True           | Double Underline                                                                    |

---

## ?CellExtraLarge

---

|                |                                                                           |
|----------------|---------------------------------------------------------------------------|
| <b>Purpose</b> | ?CellExtraLarge indicates whether the cell font attribute is Extra Large. |
| <b>Value</b>   | <b>Description</b>                                                        |
| False          | Not Extra Large                                                           |
| True           | Extra Large                                                               |

---

## ?CellFill

---

|                |                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | <p>?CellFill contains a value representing the current cell's fill style. The value of this variable is one of the numeric equivalents or enumerated types listed for the Style parameter of FillStyleEdit.</p> <p>For example, the following macro causes the computer to beep if the current cell contains a 20% shaded fill:</p> <pre>IF (?CellFill=1) BEEP ENDIF</pre> |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                 |                 |
|-----------------|-----------------|
| <b>See Also</b> | ♦ FillStyleEdit |
|-----------------|-----------------|

---

## ?CellFine

---

|                |                                                              |
|----------------|--------------------------------------------------------------|
| <b>Purpose</b> | ?CellFine indicates whether the cell font attribute is Fine. |
| <b>Value</b>   | <b>Description</b>                                           |
| False          | Not Fine                                                     |
| True           | Fine                                                         |



---

## ?CellFormula

---

**Purpose** ?CellFormula indicates whether the cell contains a formula.

| <b>Value</b> | <b>Description</b>         |
|--------------|----------------------------|
| False        | Does not contain a formula |
| True         | Contains a formula         |

---

## ?CellIgnoreWhenCalculating

---

**Purpose** ?CellIgnoreWhenCalculating contains a value representing whether the option to ignore the current cell or blocked cells when calculating formulas is active.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Do not ignore      |
| True         | Ignore             |

---

## ?CellItalics

---

**Purpose** ?CellItalics indicates whether the cell font attribute is Italics.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Italics        |
| True         | Italics            |

---

## ?CellJustification

---

**Purpose** ?CellJustification contains a value representing the justification of the current cell. The value of this variable is one of the numeric equivalents or enumerated types listed for the TableCellJustification command.

**See Also** ♦ TableCellJustification

---

## ?CellLarge

---

**Purpose** ?CellLarge indicates whether the cell font attribute is Large.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Large          |
| True         | Large              |

---

## ?CellLocked

---

**Purpose** ?CellLocked indicates whether the current cell is locked.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not locked         |
| True         | Locked             |

---

## ?CellNormal

---

**Purpose** ?CellNormal indicates whether the cell font attribute is Normal (no specified attributes).

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Normal         |
| True         | Normal             |

---

## ?CellOutline

---

**Purpose** ?CellOutline indicates whether the cell font attribute is Outline.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Outline        |
| True         | Outline            |

---

## ?CellRedline

---

**Purpose** ?CellRedline indicates whether the cell font attribute is Redline.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Redline        |
| True         | Redline            |

---

## ?CellShadow

---

**Purpose** ?CellShadow indicates whether the cell font attribute is Shadow.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Shadow         |
| True         | Shadow             |

---

## ?CellSmall

---

**Purpose** ?CellSmall indicates whether the cell font attribute is Small.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Small          |
| True         | Small              |

---

## ?CellSmallCaps

---

**Purpose** ?CellSmallCaps indicates whether the cell font attribute is Small Caps.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Small Caps     |
| True         | Small Caps         |

---

## ?CellSpecificAttribute

---

**Purpose** ?CellSpecificAttribute indicates whether the current cell contains cell-specific attributes.

| Value | Description                               |
|-------|-------------------------------------------|
| 0     | Does not contain cell-specific attributes |
| 1     | Contains cell-specific attributes         |

---

## ?CellSpecificJustification

---

**Purpose** ?CellSpecificJustification contains a value representing whether cell justification is active in the current cell.

| Value | Description           |
|-------|-----------------------|
| 0     | No cell justification |
| 1     | Cell justification    |

---

## ?CellStrikeout

---

**Purpose** ?CellStrikeout indicates whether the cell font attribute is Strikeout.

| Value | Description   |
|-------|---------------|
| False | Not Strikeout |
| True  | Strikeout     |

---

## ?CellSubscript

---

**Purpose** ?CellSubscript indicates whether the cell font attribute is Subscript.

| Value | Description   |
|-------|---------------|
| False | Not Subscript |
| True  | Subscript     |

---

## ?CellSuperscript

---

**Purpose** ?CellSuperscript indicates whether the cell font attribute is Superscript.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Superscript    |
| True         | Superscript        |

---

## ?CellUnderline

---

**Purpose** ?CellUnderline indicates whether the cell font attribute is Underline.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Underline      |
| True         | Underline          |

---

## ?CellVeryLarge

---

**Purpose** ?CellVeryLarge indicates whether the cell font attribute is Very Large.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Very Large     |
| True         | Very Large         |

---

## ?CenterCurrentPage

---

**Purpose** ?CenterCurrentPage indicates whether the contents of the current page only are set to be centered vertically.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not centered       |
| True         | Centered           |

---

**See Also** ♦ CenterCurrentPage

---

## ?CenterPages

**Purpose** ?CenterPages indicates whether the contents of the current page are centered vertically by means of the Current and Subsequent Pages setting in the Center Page(s) dialog box.

| Value | Description  |
|-------|--------------|
| False | Not centered |
| True  | Centered     |

**See Also** ♦ CenterPages

---

---

## ?ChapterNumber

**Purpose** ?ChapterNumber contains the current chapter number.

**See Also** ♦ ChapterNumber

---

---

## ?ChapterNumberMethod

**Purpose** ?ChapterNumberMethod contains a value representing the current chapter numbering method. The value of this variable is one of the numeric equivalents or enumerated types listed for ChapterNumberMethod.

For example, the following macro causes the computer to beep if the current chapter numbering method is uppercase roman numerals:

```
IF (?ChapterNumberMethod=4)
BEEP
ENDIF
```

**See Also** ♦ ChapterNumberMethod

---

---

## ?CharSpaceWidth

**Purpose** ?CharSpaceWidth contains a value representing the amount of space between letters. The value of this variable is one of the numeric equivalents or enumerated types listed for the CharacterWidth parameter of WordLetterSpacing, or a percentage of WordPerfect Optimal.

| Value | Description                 |
|-------|-----------------------------|
| 0     | Normal spacing              |
| 100   | WordPerfect optimal spacing |

---

**See Also** ♦ WordLetterSpacing

---

## ?Column

---

**Purpose** ?Column contains the number of the current text column or table column.

---

## ?ColumnBorder

---

**Purpose** ?ColumnBorder contains a value representing whether the current column setting includes a border.

| Value | Description |
|-------|-------------|
| 0     | No border   |
| 1     | Border      |

---

## ?ColumnDefine

---

**Purpose** ?ColumnDefine contains a value representing the current column type. The value of this variable is one of the numeric equivalents or enumerated types listed for the ColumnsType parameter of ColumnsDefinition.

For example, the following macro causes the computer to beep if the current column type is Parallel:

```
IF (?ColumnDefine=2)
BEEP
ENDIF
```

---

**See Also** ♦ ColumnsDefinition

---

## ?ConfirmCodesDelete

---

**Purpose** ?ConfirmCodesDelete indicates whether the environment preference Confirm Deletion of Codes, Stop Insertion Point at Hidden Codes is on or off.

| Value | Description |
|-------|-------------|
| False | Off         |
| True  | On          |

---

## ?ConfirmFormulaDelete

---

**Purpose** ?ConfirmFormulaDelete indicates whether the environment preference Confirm Deletion of Table Formulas is on or off.

| Value | Description |
|-------|-------------|
| False | Off         |
| True  | On          |

---

## ?CurrentLabel

---

**Purpose** ?CurrentLabel contains the name of the current label selection.

---

## ?CurrentParagraphLevel

---

**Purpose** ?CurrentParagraphLevel contains a value representing the level of the current outline paragraph.

---

## ?CurrentPaperName

---

**Purpose** ?CurrentPaperName contains the paper definition currently selected in the Paper Size dialog box.



---

## ?CurrentPrinter

**Purpose** ?CurrentPrinter contains the name of the current printer.

---

## ?CurrentPrinterPort

**Purpose** ?CurrentPrinterPort contains the name of the current printer port.

---

## ?CurrentPrinterType

**Purpose** ?CurrentPrinterType contains a value representing whether the current printer driver is WordPerfect or Windows.

| Value | Description |
|-------|-------------|
| 0     | WordPerfect |
| 4     | Windows     |

---

## ?CurrentSubdoc

**Purpose** ?CurrentSubdoc contains a value representing which substructure is active.

| Value | Description             |
|-------|-------------------------|
| 0     | None                    |
| 1     | Footnote                |
| 2     | Endnote                 |
| 3     | Header A                |
| 4     | Header B                |
| 5     | Footer A                |
| 6     | Footer B                |
| 7     | Watermark A             |
| 8     | Watermark B             |
| 9     | ToA Full Form           |
| 10    | Window Caption          |
| 11    | Text Box                |
| 12    | Equation Editor         |
| 13    | Comment                 |
| 14    | Endnote (for placement) |
| 15    | Envelope                |
| 16    | Style                   |

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| 17           | Equation Display   |
| 18           | Delay Codes        |
| 19           | Summary Abstract   |

---

## ?CurrentTableName

---

**Purpose** ?CurrentTableName contains the current table name, or the previous table name if the insertion point is not in a table, or 0 if there is no previous table.

---

## ?CurrentTemplate

---

**Purpose** ?CurrentTemplate contains the current template's path and filename.

---

## ?CurrentWord

---

**Purpose** ?CurrentWord contains the word at the insertion point.

---

## ?DateDay

---

**Purpose** ?DateDay contains a numeric value representing the current day of the month.

---

## ?DateMonth

---

**Purpose** ?DateMonth contains a numeric value representing the current month.

---

## ?DateWeekday

---

**Purpose**                   ?DateWeekday contains the current day of the week.

---

---

## ?DateYear

---

**Purpose**                   ?DateYear contains a numeric value representing the current year.

---

---

## ?DecimalAlignmentCharacter

---

**Purpose**                   ?DecimalAlignmentCharacter the current decimal alignment character.

---

**See Also**               ♦ DecimalAlignmentCharacter

---

---

## ?DefaultDocumentExtension

---

**Purpose**                   ?DefaultDocumentExtension contains the default extension for document filenames.

---

---

## ?DefaultTemplateExtension

---

**Purpose**                   ?DefaultTemplateExtension contains the default extension for template filenames.

---

---

## ?DefaultView

---

**Purpose**                   ?DefaultView contains a value representing the default view preference. The value of this variable is one of the numeric equivalents or enumerated types listed for the DefaultView parameter of PrefDisplaySet.

For Example:

```
IF (?DefaultView=1)
BEEP
ENDIF
```

---

## ?DisplayMode

### Purpose

?DisplayMode contains a value representing the current display mode. The value of this variable is one of the numeric equivalents or enumerated types listed for DisplayMode.

For example, the following macro causes the computer to beep if the current display mode is Graphics mode:

```
IF (?DisplayMode=2)
BEEP
ENDIF
```

---

## ?DisplayPitch

### Purpose

?DisplayPitch contains a value representing whether display pitch is set to auto or manual.

| Value | Description |
|-------|-------------|
| 0     | Automatic   |
| 1     | Manual      |

---

## ?DocBlank

### Purpose

?DocBlank contains a value representing whether the current document is blank.

| Value | Description |
|-------|-------------|
| False | Not blank   |
| True  | Blank       |

---

## ?DocChanged

### Purpose

?DocChanged contains a value representing whether the current document has been modified since the last time it was saved.

| Value | Description |
|-------|-------------|
| False | Not changed |
| True  | Changed     |

---

## ?DocNumber

**Purpose** ?DocNumber contains the current document number. Values range from 1 to 9.

---

## ?DocReadOnly

**Purpose** ?DocReadOnly contains a value representing whether the current document is read-only.

| Value | Description   |
|-------|---------------|
| False | Not read-only |
| True  | Read-only     |

---

## ?DocumentFont

**Purpose** ?DocumentFont contains the current font name.

---

## ?DocumentInformation

**Purpose** ?DocumentInformation contains the number of characters in the current document.

---

## ?DotLeader

**Purpose** ?DotLeader contains the current dot leader character.

**See Also** ♦ DotLeader

---

---

## ?EditScreen

**Purpose** ?EditScreen contains a value representing whether the current screen is an editing screen.

| Value | Description           |
|-------|-----------------------|
| False | Not an editing screen |
| True  | Editing screen        |

---

## ?Endnote

**Purpose** ?Endnote contains the number of the endnote to the left of the insertion point.

---

## ?EndnoteMinimum

**Purpose** ?EndnoteMinimum contains a measurement expression indicating the amount of endnote text to keep together.

**See Also** ♦ EndnoteMinimum

---

## ?EndnoteSpace

**Purpose** ?EndnoteSpace contains a measurement expression indicating the amount of blank space between endnotes.

---

## ?FeatureBar

**Purpose** ?FeatureBar contains a value representing which feature bar is displayed.

| Value | Description              |
|-------|--------------------------|
| 0     | No feature bar displayed |
| 35    | Comment                  |
| 91    | Footnote/Endnote         |
| 92    | Header/Footer            |
| 94    | Hypertext                |
| 109   | Graphics Box             |

|     |                           |
|-----|---------------------------|
| 124 | Mark List                 |
| 125 | Mark Table of Contents    |
| 127 | Keyboard Merge            |
| 137 | Mark Table of Authorities |
| 138 | Mark Index                |
| 143 | Outline                   |
| 167 | Mark Cross Reference      |
| 243 | Watermark                 |
| 265 | Merge Form                |
| 270 | Merge Table               |
| 271 | Merge Text                |
| 275 | Figure Edit               |
| 280 | Transcribe                |
| 290 | Table Formula             |
| 309 | TOA Full Form             |
| 327 | Delay Codes               |
| 337 | Template                  |
| 340 | Box Caption Edit          |
| 341 | Box Text Edit             |
| 342 | Macro                     |

---

## ?Fill

### Purpose

?Fill contains a value representing the current fill style. The value of this variable is one of the numeric equivalents or enumerated types listed for FillStyleEdit. This variable must be preceded by a command which specifies a fill style, such as FillStyleEdit, and followed by a closing command such as FillStyleEnd.

For example, the following macro types the name of the style specified by FillStyleEdit:

```
FillStyleEdit (Style: Fill10!)
Type (?Fill)
FillStyleEnd (State: Save!)
```

### See Also

- FillStyleEdit

---

## ?FillGradientSteps

### Purpose

?FillGradientSteps contains a value representing whether steps in a gradient are calculated automatically. This variable must be preceded by a command which specifies a fill style, such as FillStyleEdit, and followed by a closing command such as FillStyleEnd.

| Value | Description                  |
|-------|------------------------------|
| 0     | Not calculated automatically |
| 1     | Calculated automatically     |

---

## ?FillGradientType

### Purpose

?FillGradientType contains a value representing whether the current gradient fill is a linear, radial, or rectangular gradient. The value of this variable is one of the numeric equivalents or enumerated types listed for the Gradient parameter of FillSetGradient. This variable must be preceded by a command which specifies a fill style, such as FillStyleEdit, and followed by a closing command such as FillStyleEnd.

For example, the following macro causes the computer to beep if the fill style specified is a radial gradient:

```
FillStyleEdit (Style: Fill10!)
IF (?FillGradientType=1)
BEEP
ENDIF
FillStyleEnd (State: Save!)
```

### See Also

- ◆ FillSetGradient

---

## ?FillType

### Purpose

?FillType contains a value representing whether the fill is a pattern or a gradient. This variable must be preceded by either FillStyleCreate or FillStyleEdit and followed by FillStyleEnd, or be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

| Value | Description |
|-------|-------------|
| 0     | Pattern     |
| 1     | Gradient    |

---

## ?FillWPG2Pattern

### Purpose

?FillWPG2Pattern contains a value representing a predefined fill pattern. The value of this variable is one of the numeric equivalents or enumerated types listed for the Index parameter of FillSetWPG2. This variable must be preceded by either FillStyleCreate



or FillStyleEdit and followed by FillStyleEnd, or be preceded by either BoxCreate or BoxEdit and followed by BoxEnd.

For example, the following macro types a value representing the pattern specified for Button Fill:

```
FillStyleEdit (FillName: "Button Fill")
Type (?FillWPG2Pattern)
FillStyleEnd (State: Save!)
```

---

**See Also**

- ◆ FillSetWPG2

---

## ?FirstLineIndent

---

**Purpose** ?FirstLineIndent contains a measurement expression indicating the amount the first line of a new paragraph is indented.

---

## ?FirstPrintLineNum

---

**Purpose** ?FirstPrintLineNum contains the first line number to print.

---

## ?Font

---

**Purpose** ?Font contains the current font name.

---

## ?FontBold

---

**Purpose** ?FontBold contains a value representing whether the current font attribute is Bold.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Bold           |
| True         | Bold               |

---

## ?FontDoubleUnderline

---

**Purpose**

?FontDoubleUnderline contains a value representing whether the current font attribute is Double Underline.

| <b>Value</b> | <b>Description</b>   |
|--------------|----------------------|
| False        | Not Double Underline |
| True         | Double Underline     |

---

## ?FontExtraLarge

---

**Purpose**

?FontExtraLarge contains a value representing whether the current font attribute is Extra Large.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Extra Large    |
| True         | Extra Large        |

---

## ?FontFine

---

**Purpose**

?FontFine contains a value representing whether the current font attribute is Fine.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Fine           |
| True         | Fine               |

---

## ?FontItalics

---

**Purpose**

?FontItalics contains a value representing whether the current font attribute is Italics.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Italics        |
| True         | Italics            |

---

## ?FontLarge

|                |                                                                                       |                    |
|----------------|---------------------------------------------------------------------------------------|--------------------|
| <b>Purpose</b> | ?FontLarge contains a value representing whether the current font attribute is Large. |                    |
|                | <b>Value</b>                                                                          | <b>Description</b> |
|                | False                                                                                 | Not Large          |
|                | True                                                                                  | Large              |

---

## ?FontOutline

|                |                                                                                           |                    |
|----------------|-------------------------------------------------------------------------------------------|--------------------|
| <b>Purpose</b> | ?FontOutline contains a value representing whether the current font attribute is Outline. |                    |
|                | <b>Value</b>                                                                              | <b>Description</b> |
|                | False                                                                                     | Not Outline        |
|                | True                                                                                      | Outline            |

---

## ?FontRedline

|                |                                                                                           |                    |
|----------------|-------------------------------------------------------------------------------------------|--------------------|
| <b>Purpose</b> | ?FontRedline contains a value representing whether the current font attribute is Redline. |                    |
|                | <b>Value</b>                                                                              | <b>Description</b> |
|                | False                                                                                     | Not Redline        |
|                | True                                                                                      | Redline            |

---

## ?FontShadow

|                |                                                                                         |                    |
|----------------|-----------------------------------------------------------------------------------------|--------------------|
| <b>Purpose</b> | ?FontShadow contains a value representing whether the current font attribute is Shadow. |                    |
|                | <b>Value</b>                                                                            | <b>Description</b> |
|                | False                                                                                   | Not Shadow         |
|                | True                                                                                    | Shadow             |

---

## ?FontSize

**Purpose** ?FontSize contains a value representing the current font size.

---

## ?FontSmall

**Purpose** ?FontSmall contains a value representing whether the current font attribute is Small.

| Value | Description |
|-------|-------------|
| False | Not Small   |
| True  | Small       |

---

## ?FontSmallCaps

**Purpose** ?FontSmallCaps contains a value representing whether the current font attribute is Small Caps.

| Value | Description    |
|-------|----------------|
| False | Not Small Caps |
| True  | Small Caps     |

---

## ?FontStrikeout

**Purpose** ?FontStrikeout contains a value representing whether the current font attribute is Strikeout.

| Value | Description   |
|-------|---------------|
| False | Not Strikeout |
| True  | Strikeout     |

---

## ?FontSubscript

**Purpose** ?FontSubscript contains a value representing whether the current font attribute is Subscript.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Subscript      |
| True         | Subscript          |

---

## ?FontSuperscript

**Purpose** ?FontSuperscript contains a value representing whether the current font attribute is Superscript.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Superscript    |
| True         | Superscript        |

---

## ?FontUnderline

**Purpose** ?FontUnderline contains a value representing whether the current font attribute is Underline.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Underline      |
| True         | Underline          |

---

## ?FontVeryLarge

**Purpose** ?FontVeryLarge contains a value representing whether the current font attribute is Very Large.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not Very Large     |
| True         | Very Large         |

---

## ?FooterAOccur

### Purpose

?FooterAOccur contains a value representing whether Footer A prints on odd, even, or all pages. The value of this variable is one of the numeric equivalents or enumerated types listed for the Occurrence parameter of FooterA.

For example, the following macro causes the computer to beep if Footer A occurs on odd pages:

```
IF (?FooterAOccur=1)
BEEP
ENDIF
```

### See Also

♦ FooterA

---

---

## ?FooterBOccur

### Purpose

?FooterBOccur contains a value representing whether Footer B prints on odd, even, or all pages. The value of this variable is one of the numeric equivalents or enumerated types listed for the Occurrence parameter of FooterB.

For example, the following macro causes the computer to beep if Footer B occurs on even pages:

```
IF (?FooterBOccur=2)
BEEP
ENDIF
```

### See Also

♦ FooterB

---

---

## ?FooterSeparator

### Purpose

?FooterSeparator contains a measurement expression indicating the amount of blank space above a footer.

---

---

## ?Footnote

### Purpose

?Footnote contains the footnote number to the left of the insertion point.

---

---

## ?FootnoteContinued

### Purpose

?FootnoteContinued contains a value representing whether to print a continued message when a footnote is split between two pages.

| Value | Description             |
|-------|-------------------------|
| 0     | No continued message    |
| 1     | Print continued message |

---

## ?FootnoteMinimum

### Purpose

?FootnoteMinimum contains a measurement expression indicating the minimum amount of footnote text to keep together.

---

## ?FootnotePosition

### Purpose

?FootnotePosition contains a value representing whether footnotes are printed at the bottom of the page or just below the last line of text.

| Value | Description    |
|-------|----------------|
| 0     | Bottom of page |
| 1     | After text     |

---

## ?FootnoteRestartsPage

### Purpose

?FootnoteRestartsPage contains a value representing whether footnote numbering restarts on every page.

| Value | Description      |
|-------|------------------|
| 0     | Does not restart |
| 1     | Restarts         |

---

## ?FootnoteSpace

**Purpose**                   ?FootnoteSpace contains a measurement expression indicating the amount of blank space between footnotes.

---

## ?GraphicsLineBottomSpacing

**Purpose**                   ?GraphicsLineBottomSpacing contains a value representing the amount of space below a graphics line. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the amount of space below line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineBottomSpacing)
GraphicsLineEnd (State: Save!)
```

---

**See Also**                   ♦ GraphicsLineSpacing

---

## ?GraphicsLineColorBlue

**Purpose**                   ?GraphicsLineColorBlue contains a value representing the amount of blue in a graphics line color. Values range from 0 to 255, with 0 indicating no blue and 255 indicating solid blue. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the amount of blue in line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineColorBlue)
GraphicsLineEnd (State: Save!)
```

---

**See Also**                   ♦ GraphicsLineColor



---

## ?GraphicsLineColorGreen

---

**Purpose** ?GraphicsLineColorGreen contains a value representing the amount of green in a graphics line color. Values range from 0 to 255, with 0 indicating no green and 255 indicating solid green. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the amount of green in line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineColorGreen)
GraphicsLineEnd (State: Save!)
```

---

**See Also** ♦ GraphicsLineColor

---

---

## ?GraphicsLineColorRed

---

**Purpose** ?GraphicsLineColorRed contains a value representing the amount of red in a graphics line color. Values range from 0 to 255, with 0 indicating no red and 255 indicating solid red. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the amount of red in line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineColorRed)
GraphicsLineEnd (State: Save!)
```

---

**See Also** ♦ GraphicsLineColor

---

---

## ?GraphicsLineColorShading

---

**Purpose** ?GraphicsLineColorShading contains a value representing the shade of a graphics line color. Values range from 0 to 100, with 100 indicating the darkest shade of a color. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the shade of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineColorShading)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ◆ GraphicsLineColor

---

## ?GraphicsLineCount

**Purpose**

?GraphicsLineCount contains the number of graphics lines in the current document.

---

## ?GraphicsLineHorizontalAlign

**Purpose**

?GraphicsLineHorizontalAlign contains a value representing the horizontal alignment of a graphics line. The value of this variable is one of the numeric equivalents or enumerated types listed for the Position parameter of GraphicsLineHorizontalPosition. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the horizontal position of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineHorizontalAlign)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ◆ GraphicsLineHorizontalPosition

---

## ?GraphicsLineHorizontalColumn

**Purpose**

?GraphicsLineHorizontalColumn contains a column number. When the horizontal position of a graphics line is set to Between Columns, the value of this variable is the number of the column to the left of the line. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types the number of the column to the left of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineHorizontalColumn)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ◆ GraphicsLineHorizontalPosition

---

## ?GraphicsLineHorizontalPosition

---

### Purpose

?GraphicsLineHorizontalPosition contains a value representing the horizontal position of a graphics line. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the horizontal position of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineHorizontalPosition)
GraphicsLineEnd (State: Save!)
```

---

### See Also

- ♦ GraphicsLineHorizontalPosition

---

## ?GraphicsLineLength

---

### Purpose

?GraphicsLineLength contains the length of a graphics line. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the length of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineLength)
GraphicsLineEnd (State: Save!)
```

---

### See Also

- ♦ GraphicsLineLength

---

## ?GraphicsLineNum

---

### Purpose

?GraphicsLineNum contains the current graphics line number.

---

## ?GraphicsLineStyle

---

### Purpose

?GraphicsLineStyle contains a value representing a graphics line style. The value of this variable is either one of the numeric equivalents or enumerated types listed for GraphicsLineStyle, or the name of a user-created style. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the style of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineStyle)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ◆ GraphicsLineStyle

---

## ?GraphicsLineThickness

---

**Purpose**

?GraphicsLineThickness contains a value representing the thickness of a graphics line. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the thickness of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineThickness)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ◆ GraphicsLineThickness

---

## ?GraphicsLineTopSpacing

---

**Purpose**

?GraphicsLineTopSpacing contains a value representing the amount of space above a graphics line. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the amount of space above line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineTopSpacing)
GraphicsLineEnd (State: Save!)
```

---

**See Also**

- ◆ GraphicsLineSpacing

---

## ?GraphicsLineType

---

**Purpose** ?GraphicsLineType contains a value representing whether a graphics line is horizontal or vertical. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

| Value | Description     |
|-------|-----------------|
| 0     | Horizontal line |
| 1     | Vertical line   |

---

**See Also** ♦ GraphicsLineType

---

## ?GraphicsLineUseColor

---

**Purpose** ?GraphicsLineUseColor contains a value representing whether to use the current line style color or to choose another color. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

| Value | Description          |
|-------|----------------------|
| 0     | Use line style color |
| 1     | Choose a color       |

---

**See Also** ♦ GraphicsLineUseColor

---

## ?GraphicsLineVerticalAlign

---

**Purpose** ?GraphicsLineVerticalAlign contains a value representing the vertical alignment of a graphics line. The value of this variable is one of the numeric equivalents or enumerated types listed for the Position parameter of GraphicsLineVerticalPosition. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the vertical alignment of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineVerticalAlign)
GraphicsLineEnd (State: Save!)
```

---

**See Also** ♦ GraphicsLineVerticalPosition

---

## ?GraphicsLineVerticalPosition

---

### Purpose

?GraphicsLineVerticalPosition contains a value representing the vertical position of a graphics line. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either GraphicsLineCreate or GraphicsLineEdit and followed by GraphicsLineEnd.

For example, the following macro types a value representing the vertical position of line 1:

```
GraphicsLineEdit (Line: 1)
Type (?GraphicsLineVerticalPosition)
GraphicsLineEnd (State: Save!)
```

---

### See Also

- ◆ GraphicsLineVerticalPosition

---

## ?HeaderAOccur

---

### Purpose

?HeaderAOccur contains a value representing whether Header A prints on odd, even, or all pages. If the header is active, the value of this variable is one of the numeric equivalents or enumerated types listed for the Occurrence parameter of HeaderA. A value of 0 is returned if the header is not active.

For example, the following macro causes the computer to beep if Header A occurs on odd pages:

```
IF (?HeaderAOccur=1)
BEEP
ENDIF
```

---

### See Also

- ◆ HeaderA

---

## ?HeaderBOccur

---

### Purpose

?HeaderBOccur contains a value representing whether Header B prints on odd, even, or all pages. If the header is active, the value of this variable is one of the numeric equivalents or enumerated types listed for the Occurrence parameter of HeaderB. A value of 0 is returned if the header is not active.

For example, the following macro causes the computer to beep if Header B occurs on odd pages:

```
IF (?HeaderBOccur=1)
BEEP
ENDIF
```

---

**See Also**

- ♦ HeaderB

---

## ?HeaderSeparator

---

**Purpose** ?HeaderSeparator contains a measurement expression in WordPerfect units (1200ths of an inch) indicating the amount of blank space above a header.

---

**See Also** ♦ HeaderSeparationDistance

---

## ?HypertextLast

---

**Purpose** ?HypertextLast contains the filename of the file in which a hypertext link originated.

---

## ?HypertextType

---

**Purpose** ?HypertextType contains a value representing whether a hypertext link jumps to a bookmark or plays a macro.

| Value | Description |
|-------|-------------|
| 2     | Bookmark    |
| 4     | Macro       |

---

## ?Hyphenation

---

**Purpose** ?Hyphenation contains a value representing whether hyphenation is on or off.

| Value | Description     |
|-------|-----------------|
| 0     | Hyphenation off |
| 1     | Hyphenation on  |

---

**See Also** ♦ Hyphenation

---

## ?Indent

**Purpose** ?Indent contains a value representing whether the first line of a new paragraph is indented.

| Value | Description  |
|-------|--------------|
| 0     | Not indented |
| 1     | Indented     |

**See Also** ♦ Indent

---

---

## ?IndexDefinition

**Purpose** ?IndexDefinition contains a value representing the current index heading style. The value of this variable is one of the numeric equivalents or enumerated types listed for the Heading parameter of IndexDefinition.

For example, the following macro causes the computer to beep if the current index heading style is Index 2 Style:

```
IF (?IndexDefinition=30)
BEEP
ENDIF
```

**See Also** ♦ IndexDefinition

---

---

## ?Initials

**Purpose** ?Initials contains the initials of the current user. This variable is only valid if WordPerfect is running on a network.

---

---

## ?InOutline

**Purpose** ?InOutline contains a value representing whether the insertion point is in an outline.

| Value | Description       |
|-------|-------------------|
| False | Not in an outline |
| True  | In an outline     |



---

## ?InterimVersion

**Purpose** ?InterimVersion contains the interim release number of the current version of WordPerfect. For Example, the value assigned to the initial release is 0, and the value assigned to the first interim release is 1.

---

## ?Justification

**Purpose** ?Justification contains a value representing the current justification. The value of this variable is one of the numeric equivalents or enumerated types listed for Justification. For example, the following macro causes the computer to beep if justification is set to Center:

```
IF (?Justification=2)
BEEP
ENDIF
```

**See Also** ♦ Justification

---

## ?Kerning

**Purpose** ?Kerning contains a value representing whether Automatic Kerning is on or off.

| Value | Description |
|-------|-------------|
| 0     | Off         |
| 1     | On          |

**See Also** ♦ Kerning

---

## ?LabelsToDisplay

**Purpose** ?LabelsToDisplay contains a value representing whether to display labels for laser or tractor-feed printers, or both types. The value of this variable is one of the numeric equivalents or enumerated types listed for the LabelsDisplay parameter of PrefLocationOfPrinterFiles.

For Example:

```
IF (?LabelsToDisplay=1)
BEEP
ENDIF
```

---

## ?Language

|                 |                                                                                                                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ?Language contains a value representing the current language. The value of this variable is a WordPerfect language code. For example, if the current language is U.S. English, the value is US. |
| <b>See Also</b> | ♦ Language                                                                                                                                                                                      |

---

## ?LeftChar

|                |                                                                                  |
|----------------|----------------------------------------------------------------------------------|
| <b>Purpose</b> | ?LeftChar contains the character immediately to the left of the insertion point. |
|----------------|----------------------------------------------------------------------------------|

---

## ?LeftCode

|                |                                                                                                                                                                    |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | ?LeftCode contains a value representing the code immediately to the left of the insertion point. The value of this variable is the numeric equivalent of the code. |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## ?LeftHyphZone

|                 |                                                                                                                                                                          |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b>  | ?LeftHyphZone contains a value representing the lower limit of a hyphenation zone. The value of this variable is a numeric expression indicating a percentage of a word. |
| <b>See Also</b> | ♦ HyphenationZoneLeft                                                                                                                                                    |

---

## ?LeftMarginAdj

**Purpose** ?LeftMarginAdj contains a measurement expression indicating the amount of left margin adjustment for the current paragraph.

**See Also** ♦ MarginAdjustLeft

---

## ?Line

**Purpose** ?Line contains a measurement expression indicating the vertical position of the insertion point relative to the top of the page.

---

## ?LineHeight

**Purpose** ?LineHeight contains a value representing whether line height is fixed or set automatically.

| Value | Description |
|-------|-------------|
| 0     | Fixed       |
| 1     | Automatic   |

**See Also** ♦ LineHeight

---

## ?LineNumber

**Purpose** ?LineNumber contains the starting line number for the current document.

---

## ?LineNumbering

**Purpose** ?LineNumbering contains a value representing whether Line Numbering is on or off.

| Value | Description        |
|-------|--------------------|
| False | Line numbering off |
| True  | Line numbering on  |

---

**See Also**

- ◆ LineNumbering

---

## ?LineNumberingInterval

**Purpose**

?LineNumberingInterval contains a number indicating the interval between numbered lines.

**See Also**

- ◆ LineNumbering

---

## ?LineNumberMethod

**Purpose**

?LineNumberMethod contains a value representing the line numbering method. The value of this variable is one of the numeric equivalents or enumerated types listed for LineNumberingMethod.

For example, the following macro causes the computer to beep if the line numbering method is uppercase roman numerals:

```
IF (?LineNumberMethod=4)
BEEP
ENDIF
```

---

## ?LineNumCountBlankLines

**Purpose**

?LineNumCountBlankLines contains a value representing whether Line Numbering counts blank lines.

| <b>Value</b> | <b>Description</b>       |
|--------------|--------------------------|
| False        | Do not count blank lines |
| True         | Count blank lines        |

**See Also**

- ◆ LineNumbering

---

## ?LineNumNewspaperCols

---

**Purpose** ?LineNumNewspaperCols contains a value representing whether Line Numbering numbers newspaper columns.

| Value | Description                     |
|-------|---------------------------------|
| False | Do not number newspaper columns |
| True  | Number newspaper columns        |

---

**See Also** ♦ LineNumbering

---

## ?LineNumPosition

---

**Purpose** ?LineNumPosition returns in WordPerfect units the position of line numbers relative to the left edge of the page or margin, depending on user selection in the Line Numbering dialog box.

---

**See Also** ♦ LineNumbering

---

## ?LineSegmentBlue

---

**Purpose** ?LineSegmentBlue contains a value representing the amount of blue in a table line segment color. Values range from 0 to 255, with 0 indicating no blue and 255 indicating solid blue. This variable must be preceded by either LineStyleCreate or LineStyleEdit and followed by LineStyleEnd, and be preceded by either LineSegmentCreate or LineSegmentEdit and followed by LineSegmentEnd.

For example, the following macro types a value representing the amount of blue in a specified line segment:

```
LineStyleEdit (Style: "My Style")
LineSegmentEdit (Position: 0)
Type (?LineSegmentBlue)
LineSegmentEnd (State: Save!)
LineStyleEnd (State: Save!)
```

---

**See Also** ♦ LineSegmentColor

---

## ?LineStyleCount

---

### Purpose

?LineStyleCount contains the number of line segments in a line style. This variable must be preceded by either LineStyleCreate or LineStyleEdit and followed by LineStyleEnd.

For Example:

```
LineStyleEdit (Style: "My Style")
Type (?LineStyleCount)
LineStyleEnd (State: Save!)
```

---

## ?LineStyleGreen

---

### Purpose

?LineStyleGreen contains a value representing the amount of green in a line segment color. Values range from 0 to 255, with 0 indicating no green and 255 indicating solid green. This variable must be preceded by either LineStyleCreate or LineStyleEdit and followed by LineStyleEnd, and preceded by either LineSegmentCreate or LineSegmentEdit and followed by LineSegmentEnd.

For example, the following macro types a value representing the amount of green in a specified line segment:

```
LineStyleEdit (Style: "My Style")
LineSegmentEdit (Position: 0)
Type (?LineStyleGreen)
LineSegmentEnd (State: Save!)
LineStyleEnd (State: Save!)
```

---

### See Also

- ◆ LineSegmentColor

---

## ?LineStyleRed

---

### Purpose

?LineStyleRed contains a value representing the amount of red in a line segment color. Values range from 0 to 255, with 0 indicating no red and 255 indicating solid red. This variable must be preceded by either LineStyleCreate or LineStyleEdit and followed by LineStyleEnd, and be preceded by either LineSegmentCreate or LineSegmentEdit and followed by LineSegmentEnd.

For example, the following macro types a value representing the amount of red in a specified line segment:

```
LineStyleEdit (State: "My Style")
LineSegmentEdit (Position: 0)
Type (?LineStyleRed)
```

LineSegmentEnd (State: Save!)  
LineStyleEnd (State: Save!)

---

**See Also**

- ◆ LineSegmentColor

---

## ?LineSegmentSetUserData

---

**Purpose**

?LineSegmentSetUserData contains a measurement expression indicating the dash length in the current custom line pattern. This variable must be preceded by LineSegmentSetUser and followed by LineSegmentSetUserEnd.

For example, the following macro types a value representing the dash length in a specified line segment:

```
LineSegmentSetUser (OnOffs: 2)
Type (?LineSegmentSetUserData)
LineSegmentSetUserEnd ()
```

---

**See Also**

- ◆ LineSegmentSetUserData

---

## ?LineSegmentShading

---

**Purpose**

?LineSegmentShading contains a value representing the shade of a line segment color. Values range from 0 to 100, with 100 indicating the darkest shade of a color. This variable must be preceded by either LineStyleCreate or LineStyleEdit and followed by LineStyleEnd, and be preceded by either LineSegmentCreate or LineSegmentEdit and followed by LineSegmentEnd.

For example, the following macro types a value representing the shade of a specified line segment:

```
LineSegmentEdit (Position: 0)
Type (?LineSegmentShading)
LineSegmentEnd (State: Save!)
LineStyleEnd (State: Save!)
```

---

**See Also**

- ◆ LineSegmentColor

---

## ?LineSegmentSpacing

---

### Purpose

?LineSegmentSpacing contains a value representing the amount of space between one stacked line segment and the next. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either `LineStyleCreate` or `LineStyleEdit` and followed by `LineStyleEnd`, and be preceded by either `LineSegmentCreate` or `LineSegmentEdit` and followed by `LineSegmentEnd`.

For example, the following macro types a value representing the amount of space between specified stacked line segments:

```
LineStyleEdit (Style: "My Style")
LineSegmentEdit (Position: 0)
Type (?LineSegmentSpacing)
LineSegmentEnd (State: Save!)
LineStyleEnd (State: Save!)
```

---

### See Also

- ◆ `LineSegmentSpacing`

---

## ?LineSegmentWidth

---

### Purpose

?LineSegmentWidth contains the thickness of a line segment. The value of this variable is a measurement expression using the current unit of measure. This variable must be preceded by either `LineStyleCreate` or `LineStyleEdit` and followed by `LineStyleEnd`, and be preceded by either `LineSegmentCreate` or `LineSegmentEdit` and followed by `LineSegmentEnd`.

For example, the following macro types a value representing the width of a specified line segment:

```
LineStyleEdit (Style: "My Style")
LineSegmentEdit (Position: 0)
Type (?LineSegmentWidth)
LineSegmentEnd (State: Save!)
LineStyleEnd (State: Save!)
```

---

### See Also

- ◆ `LineSegmentWidth`

---

## ?LineSegmentWPG2Pattern

---

### Purpose

?LineSegmentWPG2Pattern contains a value representing a predefined line segment pattern. The value of this variable is one of the numeric equivalents or enumerated types listed for the `LineSegmentSetWPG2`. This variable must be preceded by either



LineStyleCreate or LineStyleEdit and followed by LineStyleEnd, and be preceded by either LineSegmentCreate or LineSegmentEdit and followed by LineSegmentEnd.

For example, the following macro types a value representing the pattern of a specified line segment:

```
LineStyleEdit (Style: "My Style")
LineSegmentEdit (Position: 0)
Type (?LineSegmentWPG2Pattern)
LineSegmentEnd (State: Save!)
LineStyleEnd (State: Save!)
```

---

**See Also** ♦ LineSegmentSetWPG2

---

## ?LineSpacing

---

**Purpose** ?LineSpacing contains a measurement expression indicating the amount of space between lines.

---

**See Also** ♦ LineSpacing

---

## ?ListDefinition

---

**Purpose** ?ListDefinition contains the current list name.

---

**See Also** ♦ ListDefinition

---

## ?MacroExecute

---

**Purpose** ?MacroExecute contains a value representing whether a macro is playing.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not playing        |
| True         | Playing            |

---

## ?MacroPlaying

**Purpose** ?MacroPlaying contains a value representing whether a macro is playing.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not playing        |
| True         | Playing            |

---

## ?MacroRecord

**Purpose** ?MacroRecord contains a value representing whether a macro is recording.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not recording      |
| True         | Recording          |

---

## ?MajorVersion

**Purpose** ?MajorVersion contains the major version number of the current WordPerfect version.

---

## ?MarginBottom

**Purpose** ?MarginBottom contains a value representing the size of the bottom margin. The value of this variable is a measurement expression in WordPerfect units (1200ths of an inch).

**See Also** ♦ MarginBottom

---

## ?MarginLeft

**Purpose** ?MarginLeft contains a value representing the size of the left margin. The value of this variable is a measurement expression in WordPerfect units (1200ths of an inch).

**See Also** ♦ MarginLeft

---

## ?MarginRight

**Purpose** ?MarginRight contains a value representing the size of the right margin. The value of this variable is a measurement expression in WordPerfect units (1200ths of an inch).

**See Also** ♦ MarginRight

---

## ?MarginTop

**Purpose** ?MarginTop contains a value representing the size of the top margin. The value of this variable is a measurement expression in WordPerfect units (1200ths of an inch).

**See Also** ♦ MarginTop

---

## ?MergeDone

**Purpose** ?MergeDone contains a value representing whether a merge, executed while the macro is running, has finished.

| Value | Description  |
|-------|--------------|
| False | Not finished |
| True  | Finished     |

---

## ?MinorVersion

**Purpose** ?MinorVersion contains the minor version number of the current WordPerfect version.

---

## ?Name

**Purpose** ?Name contains the current filename.

---

## ?Network

**Purpose** ?Network contains a value representing whether WordPerfect is running on a network.

| Value | Description      |
|-------|------------------|
| False | Not on a network |
| True  | On a network     |

---

## ?NotFound

**Purpose** ?NotFound contains a value representing whether the object of the last search was not found.

| Value | Description |
|-------|-------------|
| False | Found       |
| True  | Not found   |

---

## ?Overstrike

**Purpose** ?Overstrike contains the current overstrike characters.

**See Also** ♦ OverstrikeCreate

---

## ?Page

**Purpose** ?Page contains the current page number.

**See Also** ♦ PageNumber

---

## ?PageBorder

**Purpose** ?PageBorder contains a value representing whether the current page has a border.

| Value | Description |
|-------|-------------|
| 0     | No border   |
| 1     | Border      |

**See Also** ♦ PageBorderDlg

---

---

## ?PageNumberFormat

**Purpose** ?PageNumberFormat contains the text included in the current page numbering format.

**See Also** ♦ PageNumberFormat

---

---

## ?PageNumberingOn

**Purpose** ?PageNumberingOn contains a value representing whether page numbering is on in the current document.

| Value | Description       |
|-------|-------------------|
| 0     | No page numbering |
| 1     | Page numbering on |

---

## ?PageNumberingValue

**Purpose** ?PageNumberingValue contains the new page number specified in Page Settings in the Numbering Value dialog box.

---

---

## ?PageNumberMethod

---

### Purpose

?PageNumberMethod contains a value representing the current page numbering method. The value of this variable is one of the numeric equivalents or enumerated types listed for PageNumberMethod.

For example, the following macro causes the computer to beep if the current page numbering method is numbers:

```
IF (?PageNumberMethod=0)
BEEP
ENDIF
```

---

### See Also

- ♦ PageNumberMethod

---

## ?PageNumberPosition

---

### Purpose

?PageNumberPosition contains a value representing the current page number position on the page. The value of this variable is one of the numeric equivalents or enumerated types listed for the Position parameter of PageNumberPosition.

For example, the following macro causes the computer to beep if the current page number position is top left:

```
IF (?PageNumberPosition=1)
BEEP
ENDIF
```

---

### See Also

- ♦ PageNumberPosition

---

## ?PageSuppress

---

### Purpose

?PageSuppress contains a value representing the items to suppress on the current page, as set in the Suppress dialog box. If two or more items are suppressed, the value returned is the sum of the items' values.

| Value | Description                                        |
|-------|----------------------------------------------------|
| 0     | None                                               |
| 1     | Page numbering                                     |
| 2     | Print page number at bottom center on current page |
| 4     | Header A                                           |
| 8     | Header B                                           |
| 16    | Footer A                                           |
| 32    | Footer B                                           |

|     |             |
|-----|-------------|
| 64  | Watermark A |
| 128 | Watermark B |
| 253 | All         |

---

**See Also**

- ♦ PageSuppressDlg

---

## ?PaperLength

---

**Purpose**

?PaperLength contains a measurement expression indicating the current paper length in WordPerfect units of measure.

---

## ?PaperSizeType

---

**Purpose**

?PaperSizeType contains a value representing paper type of the paper definition selected in the Paper Size dialog box. The value of this variable is one of the numeric equivalents or enumerated types listed for the Type parameter of PaperSizeAdd.

For example, the following macro causes the computer to beep if a bond paper type is selected:

```
IF (?PaperSizeType=2)
BEEP
ENDIF
```

---

**See Also**

- ♦ PaperSizeDlg ♦ PaperSizeAdd

---

## ?PaperWidth

---

**Purpose**

?PaperWidth contains a measurement expression indicating the current paper width in WordPerfect units of measure.

---

## ?ParagraphBorder

---

**Purpose**

?ParagraphBorder contains a value representing whether a border style is selected in the Paragraph Border dialog box.

| Value | Description              |
|-------|--------------------------|
| 0     | No border style selected |
| 1     | Border style selected    |

---

**See Also** ♦ ParagraphBorderDlg

---

## ?ParagraphNumber

**Purpose** ?ParagraphNumber contains a value indicating the paragraph number if the paragraph outline style is active. If the paragraph outline style is not active, the value returned is 0.

---

## ?ParagraphNumberMethod

**Purpose** ?ParagraphNumberMethod contains a value indicating the paragraph numbering method specified for an outline when a custom number is defined in the Edit Outline Definition dialog box. The value of this variable is one of the numeric equivalents or enumerated types listed for PageNumberMethod.

For example, the following macro causes the computer to beep if the paragraph numbering method is numbers:

```
IF (?ParagraphNumberMethod=0)
BEEP
ENDIF
```

---

**See Also** ♦ PageNumberMethod

---

## ?Path

**Purpose** ?Path contains the current file path, but does not include the filename.

---

## ?PathBackup

**Purpose** ?PathBackup contains the directory specified in File Preferences for backup files.



---

## ?PathCurrent

**Purpose** ?PathCurrent contains the current directory.

---

---

## ?PathDatabase

**Purpose** ?PathDatabase contains the directory specified for database files in the File Preferences dialog box.

---

---

## ?PathDatabaseSupplemental

**Purpose** ?PathDatabaseSupplemental contains the directory specified for supplemental database files in the File Preferences dialog box.

---

---

## ?PathDocument

**Purpose** ?PathDocument contains the directory specified for documents in the File Preferences dialog box.

---

---

## ?PathGraphics

**Purpose** ?PathGraphics contains the directory specified for graphics files in the File Preferences dialog box.

---

---

## ?PathGraphicsSupplemental

**Purpose** ?PathGraphicsSupplemental contains the directory specified for supplemental graphics files in the File Preferences dialog box.

---

---

## ?PathHyphenation

**Purpose** ?PathHyphenation contains the directory specified for Hyphenation files in the File Preferences dialog box.

---

## ?PathLabels

**Purpose** ?PathLabels contains the directory and filename specified for labels files in the File Preferences dialog box.

---

## ?PathMacros

**Purpose** ?PathMacros contains the directory specified for macros files in the File Preferences dialog box.

---

## ?PathMacrosSupplemental

**Purpose** ?PathMacrosSupplemental contains the directory specified for supplemental macros files in the File Preferences dialog box.

---

## ?PathPrinters

**Purpose** ?PathPrinters contains the directory specified for printer files in the File Preferences dialog box.

---

## ?PathPrintersSupplemental

**Purpose** ?PathPrintersSupplemental contains the directory specified for supplemental printer files in the File Preferences dialog box.

---

---

## ?PathProgram

---

**Purpose** ?PathProgram contains the WordPerfect program directory (where WPWIN60.EXE is located).

---

## ?PathShared

---

**Purpose** ?PathShared contains the directory where WordPerfect shared code files are located.

---

## ?PathSpreadsheet

---

**Purpose** ?PathSpreadsheet contains the directory specified for spreadsheet files in the File Preferences dialog box.

---

## ?PathSpreadsheetSupplemental

---

**Purpose** ?PathSpreadsheetSupplemental contains the directory specified for supplemental spreadsheet files in the File Preferences dialog box.

---

## ?PathTemplate

---

**Purpose** ?PathTemplate contains the directory specified for template files in the File Preferences dialog box.

---

## ?PathTemplateSupplemental

---

**Purpose** ?PathTemplateSupplemental contains the directory specified for additional template files in the File Preferences dialog box.

---

---

## ?PauseKey

**Purpose** ?PauseKey contains a value representing the keystroke specified to resume a paused macro.

---

## ?Position

**Purpose** ?Position contains a value representing the horizontal position of the insertion point. The value of this variable is a measurement expression given in WordPerfect units of measure (1/1200").

---

## ?PowerBarActive

**Purpose** ?PowerBarActive indicates whether the power bar is active in the current document.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Not active         |
| True         | Active             |

---

## ?Random

**Purpose** ?Random contains a random number between 0 and 32767.

---

## ?RedlineCharacter

**Purpose** ?RedlineCharacter contains a value representing the character used to mark redline text. The value of this variable is the character's ASCII numeric equivalent.

---

**See Also** ♦ RedlineCharacter

---

---

## ?RedlineMethod

---

### Purpose

?RedlineMethod contains a value representing a redline method for the current document. The value of this variable is one of the numeric equivalents or enumerated types listed for RedlineMethod.

For example, the following macro causes the computer to beep if the current redline method is printer dependent:

```
IF (?RedlineMethod=0)
BEEP
ENDIF
```

---

### See Also

♦ RedlineMethod

---

---

## ?RevealCodesActive

---

### Purpose

?RevealCodesActive contains a value representing whether Reveal Codes is active.

| Value | Description |
|-------|-------------|
| False | Not active  |
| True  | Active      |

---

## ?RightChar

---

### Purpose

?RightChar contains the character at the insertion point.

---

---

## ?RightCode

---

### Purpose

?RightCode contains a value representing the code at the insertion point. The value of this variable is the numeric equivalent of the code.

---

---

## ?RightHyphZone

**Purpose** ?RightHyphZone contains a value representing the upper limit of a hyphenation zone. The value of this variable is a numeric expression indicating a percentage of a word.

**See Also** ♦ HyphenationZoneRight

---

---

## ?RightMarginAdj

**Purpose** ?RightMarginAdj contains a measurement expression indicating the amount of right margin adjustment in WordPerfect units of measure for the current paragraph.

**See Also** ♦ MarginAdjustRight

---

---

## ?Row

**Purpose** ?Row contains the current row number in a table.

---

---

## ?RowHeader

**Purpose** ?RowHeader contains a value representing whether the current table row is a header row.

| Value | Description  |
|-------|--------------|
| 0     | Not a header |
| 1     | Header       |

---

## ?RowHeight

**Purpose** ?RowHeight contains a measurement expression in WordPerfect units of measure indicating the current table row height.

---

---

## ?RulerBarActive

---

### Purpose

?RulerBarActive contains a value representing whether the ruler bar is active in the current document.

| Value | Description |
|-------|-------------|
| False | Not active  |
| True  | Active      |

---

## ?SaveWorkspace

---

### Purpose

?SaveWorkspace contains a value representing the current Save Workspace environment preference setting.

| Value | Environment Preferences Setting |
|-------|---------------------------------|
| 0     | Always                          |
| 1     | Never                           |
| 2     | Prompt on Exit                  |

For Example, the following macro will beep if Save Workspace is set to Never in the Environment Preferences dialog box.

```
IF (?SaveWorkspace=1)
BEEP
ENDIF
```

---

## ?SecondaryPageNumber

---

### Purpose

?SecondaryPageNumber contains the current secondary page number.

### See Also

- ♦ SecondaryPageNumber

---

## ?SecondaryPageNumberMethod

---

### Purpose

?SecondaryPageNumberMethod contains a value representing the secondary page numbering method for the current document. The value of this variable is one of the numeric equivalents or enumerated types listed for SecondaryPageNumberMethod.

For example, the following macro causes the computer to beep if the secondary page numbering method is lowercase letters:

```
IF (?SecondaryPageNumberMethod=1)
BEEP
ENDIF
```

---

**See Also** ♦ SecondaryPageNumberMethod

---

## ?SelectedText

**Purpose** ?SelectedText contains the currently selected text. This variable cannot type out the selected text, but it can be used in an expression designed to evaluate the text or to assign it to another variable.

For example, the following macro beeps if the currently selected text is "Continued...":

```
IF(?SelectedText="Continued...")
BEEP
ENDIF
```

---

## ?SelectedTextSize

**Purpose** ?SelectedTextSize contains a value representing the number of bytes in selected text.

---

## ?ShowBoxes

**Purpose** ?ShowBoxes contains a value representing whether Display Preferences hide or show graphics.

| Value | Description   |
|-------|---------------|
| 0     | Hide graphics |
| 1     | Show graphics |

---

**See Also** ♦ ShowBoxes



---

## ?SortGetAction

---

**Purpose** ?SortGetAction contains a value representing whether Select Without Sorting is on or off.

| Value | Description |
|-------|-------------|
| 0     | Off         |
| 1     | On          |

---

**See Also** ♦ SortAction

---

## ?SortGetCase

---

**Purpose** ?SortGetCase contains a value representing whether the Uppercase First sort option is on or off.

| Value | Description |
|-------|-------------|
| 0     | Off         |
| 1     | On          |

---

**See Also** ♦ SortCaseOrder

---

## ?SpaceExpansion

---

**Purpose** ?SpaceExpansion contains a numeric expression representing, as a percentage, the current compression limit for word spacing justification.

---

**See Also** ♦ WordSpacingJustificationLimits

---

## ?SpacingBelowPar

---

**Purpose** ?SpacingBelowPar contains a measurement expression indicating the amount of blank space between paragraphs. The default is 1.0, which is equal to the font point size plus 2 points.

---

**See Also** ♦ ParagraphSpacing

---

## ?StartingLineNumber

---

**Purpose** ?StartingLineNumber contains the first number to use when numbering lines of text.

**See Also** ♦ LineNumbering

---

---

## ?StatusBarActive

---

**Purpose** ?StatusBarActive contains a value representing whether the status bar is active in the current document.

| Value | Description |
|-------|-------------|
| False | Not Active  |
| True  | Active      |

---

## ?Substructure

---

**Purpose** ?Substructure contains a value representing whether a substructure is active.

| Value | Description             |
|-------|-------------------------|
| False | No substructure active. |
| True  | Substructure active     |

---

## ?TableDefaultLine

---

**Purpose** ?TableDefaultLine contains a value representing the default line style of the current table. The value of this variable is one of the numeric equivalents or enumerated types listed for TableDefaultLine.

For Example:

```
IF (?TableDefaultLine=0)
BEEP
ENDIF
```

---

## ?TableGuides

---

**Purpose** ?TableGuides contains a value representing whether the Table Cell Guides display preference is on or off.

| Value | Description |
|-------|-------------|
| 0     | Off         |
| 1     | On          |

**See Also** ♦ TableGuides

---

---

## ?TableInTable

---

**Purpose** ?TableInTable contains a value representing whether the insertion point is in a table.

| Value | Description    |
|-------|----------------|
| 0     | Not in a table |
| 1     | In a table     |

---

## ?TableLastCell

---

**Purpose** ?TableLastCell contains a value representing whether the current cell is the last cell in a table.

| Value | Description   |
|-------|---------------|
| 0     | Not last cell |
| 1     | Last cell     |

---

## ?TabSettings

---

**Purpose** ?TabSettings contains a value representing whether tabs are positioned from the left edge of the paper or relative to the left margin.

| Value | Description        |
|-------|--------------------|
| 0     | Left edge of paper |
| 1     | Left margin        |

---

**See Also**

- ◆ TabSet

---

---

## ?TemplateFile

---

**Purpose**

?TemplateFile contains the path and filename of the default template.

---

---

## ?TemplateFileSupplemental

---

**Purpose**

?TemplateFileSupplemental contains the path and filename of the current supplemental template.

---

---

## ?TextColorBlue

---

**Purpose**

?TextColorBlue contains a value representing the amount of blue in the current text color. Values range from 0 to 255.

---

**See Also**

- ◆ TextColor

---

---

## ?TextColorGreen

---

**Purpose**

?TextColorGreen contains a value representing the amount of green in the current text color. Values range from 0 to 255.

---

**See Also**

- ◆ TextColor

---

---

## ?TextColorRed

---

**Purpose**

?TextColorRed contains a value representing the amount of red in the current text color. Values range from 0 to 255.

---

---

**See Also**

- ♦ `TextColor`

---

## ?`TextColumnNumber`

---

**Purpose**

?`TextColumnNumber` contains the number of the text column containing the insertion point.

---

## ?`TextColumns`

---

**Purpose**

?`TextColumns` contains the number of text columns in the current document.

---

## ?`TextShade`

---

**Purpose**

?`TextShade` contains a value representing the shade of the current text color. The value of this variable is a percentage, with 100 indicating the darkest shade of a color.

---

**See Also**

- ♦ `TextColor`

---

## ?`ThousandsSeparatorChar`

---

**Purpose**

?`ThousandsSeparatorChar` contains the character that appears in numbers greater than or equal to 1000 to separate each three digits.

---

**See Also**

- ♦ `ThousandsSeparator`

---

## ?`TimeHour`

---

**Purpose**

?`TimeHour` contains a value representing the current hour on a 24 hour clock.

---

## ?TimeHundredth

**Purpose** ?TimeHundredth contains a value representing the current hundredth of a second.

---

---

## ?TimeMinute

**Purpose** ?TimeMinute contains a value representing the current minute.

---

---

## ?TimeSecond

**Purpose** ?TimeSecond contains a value representing the current second.

---

---

## ?ToADefinition

**Purpose** ?ToADefinition contains the current table of authorities section name.

**See Also** ♦ ToADefinition

---

---

## ?ToAShortFormMarks

**Purpose** ?ToAShortFormMarks contains the name of the last table of authorities short form marked.

**See Also** ♦ ToAMarkShortForm

---

---

## ?ToAShortForms

**Purpose** ?ToAShortForms contains the name of the last short form encountered in a table of authorities.

---

## ?ToCDefinition

**Purpose** ?ToCDefinition contains the number of levels defined for the current table of contents.

**See Also** ♦ ToCDefine

---

## ?TypeoverActive

**Purpose** ?TypeoverActive contains a value representing whether Typeover is active.

| Value | Description |
|-------|-------------|
| False | Not active  |
| True  | Active      |

**See Also** ♦ Typeover

---

## ?UndeleteBuffer

**Purpose** ?UndeleteBuffer contains the number of buffers available to undelete.

---

## ?UnderlineSpaces

**Purpose** ?UnderlineSpaces contains a value representing whether blank spaces are underlined when underline is on.

| Value | Description             |
|-------|-------------------------|
| 0     | Do not underline spaces |
| 1     | Underline spaces        |

---

**See Also**

- ♦ UnderlineSpaces

---

## ?UnderlineTabs

**Purpose**

?UnderlineTabs contains a value representing whether tabs are underlined when underline is on.

| Value | Description           |
|-------|-----------------------|
| 0     | Do not underline tabs |
| 1     | Underline tabs        |

---

**See Also**

- ♦ UnderlineTabs

---

## ?UnprintableBottomMargin

**Purpose**

?UnprintableBottomMargin contains a measurement expression indicating the size of the unprintable bottom margin in WordPerfect units.

---

## ?UnprintableLeftMargin

**Purpose**

?UnprintableLeftMargin contains a measurement expression indicating the size of the unprintable left margin in WordPerfect units.

---

## ?UnprintableRightMargin

**Purpose**

?UnprintableRightMargin contains a measurement expression indicating the size of the unprintable right margin in WordPerfect units.



---

## ?UnprintableTopMargin

---

**Purpose** ?UnprintableTopMargin contains a measurement expression indicating the size of the unprintable top margin in WordPerfect units.

---

## ?UpdateQuickList

---

**Purpose** ?UpdateQuickList contains a value representing whether the Update QuickList with Changes file preference is on or off.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Off                |
| True         | On                 |

---

## ?UseDefaultDocExt

---

**Purpose** ?UseDefaultDocExt contains a value representing whether Use Default Extension on Open and Save is on in File Preferences.

| <b>Value</b> | <b>Description</b> |
|--------------|--------------------|
| False        | Off                |
| True         | On                 |

---

## ?UserColorBlue

---

**Purpose** ?UserColorBlue contains a value representing the amount of blue in the current user-defined color. Values range from 0 to 255.

---

## ?UserColorGreen

---

**Purpose** ?UserColorGreen contains a value representing the amount of green in the current user-defined color. Values range from 0 to 255.

---

## ?UserColorRed

---

**Purpose** ?UserColorRed contains a value representing the amount of red in the current user-defined color. Values range from 0 to 255.

---

---

## ?UserInitials

---

**Purpose** ?UserInitials contains the user initials specified in Environment Preferences.

---

---

## ?UserName

---

**Purpose** ?UserName contains the user name specified in Environment Preferences.

---

---

## ?VolumeNumber

---

**Purpose** ?VolumeNumber contains the current volume number.

---

**See Also** ♦ VolumeNumber

---

---

## ?VolumeNumberMethod

---

**Purpose** ?VolumeNumberMethod contains a value representing the current volume numbering method. The value of this variable is one of the numeric equivalents or enumerated types listed for VolumeNumberMethod.

For example, the following macro causes the computer to beep if the volume numbering method is numbers:

```
IF (?VolumeNumberMethod=0)
 BEEP
ENDIF
```

---

**See Also** ♦ VolumeNumberMethod

---

---

## ?WatermarkAOccur

---

### Purpose

?WatermarkAOccur contains a value representing whether Watermark A occurs on odd, even, or all pages. The value of this variable is one of the numeric equivalents or enumerated types listed for the Occurrence parameter of WatermarkA.

For example, the following macro causes the computer to beep if Watermark A occurs on all pages:

```
IF (?WatermarkA=3)
BEEP
ENDIF
```

---

### See Also

- ♦ WatermarkA

---

## ?WatermarkBOccur

---

### Purpose

?WatermarkBOccur contains a value representing whether Watermark B occurs on odd, even, or all pages. The value of this variable is one of the numeric equivalents or enumerated types listed for the Occurrence parameter of WatermarkB.

For example, the following macro causes the computer to beep if Watermark B occurs on even pages:

```
IF (?WatermarkB=2)
BEEP
ENDIF
```

---

### See Also

- ♦ WatermarkB

---

## ?WidowOrphan

---

### Purpose

?WidowOrphan indicates whether Widow/Orphan is on or off.

| Value | Description |
|-------|-------------|
| False | Off         |
| True  | On          |

---

### See Also

- ♦ WidowOrphan

---

## ?WritingToolsDisabledState

---

**Purpose** ?WritingToolsDisabledState contains a value representing whether Disable Writing Tools is on or off.

| Value | Description |
|-------|-------------|
| 0     | Off         |
| 1     | On          |

---

**See Also** ♦ WritingToolsDisable

---

---

## ?Zoom

---

**Purpose** ?Zoom contains a percentage indicating the size of the current view.

---

**See Also** ♦ ZoomDlg

---

# Coaches Reference





A Coach takes the user through some of the product's most common tasks step-by-step, providing instructions and prompts as needed. The Coach can either show the user how to perform the desired task or it can tell the user what to do and then monitor the user's actions to make sure no mistakes are made.

The Coach Support Routines allow Coaches to be written with macros for products that use Shared Code 2.x.

The support routines provide three types of functions:

### **Prompting**

CoachMessageBox displays options for the user to select. CoachPrompt displays helps and hints.

### **Animating**

CoachAnimate shows the user how to perform certain actions. The Coach can: 1) move the mouse pointer, 2) send mouse clicks, and 3) send keystrokes to the application.

### **Filtering**

CoachFilterAdd defines a filter to monitor mouse movements, keystrokes, and tokens. CoachFilterEnable activates the filter to compare defined events with current events. For example, a filter may wait for a left mouse click in the OK button of a particular dialog, then move to the next step. If the user makes a mistake, the Coach might display a message box or prompt.

CoachFilterDisable deactivates a filter and CoachFilterDestroy destroys a filter.

There are three types of filters 1) mouse, 2) keyboard, and 3) command. Mouse and keyboard filters are *interface filters*.

### **Named Region**

A named region is a specific window or item on the screen, such as "WordPerfect.EditWindow" and "WordPerfect.OpenDlg.OpenBtn". See *Named Region* in the Mouse String section below.

### **Writing the Coach**

A Coach generally follows these steps:

- 1** Pause keyboard and mouse input until a Coach enables it.
- 2** Define filter.
- 3** Instruct the user how to perform a task with prompts, message boxes, and animation.
- 4** Enable filter.
- 5** Wait for user input.
- 6** Respond to user input with a callback procedure.
- 7** Disable filter.

- 8 Destroy filter.
- 9 Turn on keyboard and mouse input.

Define all filters at the beginning of a Coach macro, then enable each one as it is needed. Repeat steps 2 through 8 as many times as necessary.

### Quitting the Coach

Press Alt+Esc to quit the Coach.

---

## The Coach Dialog Box

Coaches have a .WCH extension and are stored in the macro directory.

When the user chooses Coach from the Help menu, the descriptive names of all .WCH files in the macros directory are displayed in the Coach dialog box. When a Coach name is selected, the information from the Abstract field of the document summary is displayed at the bottom of the Coach dialog box. If the Coach does not have a descriptive name, the filename is displayed.

---

## Coach Commands

CoachSetDialogFilter and CoachRemoveDialogFilter are available in Shared Code 2.1 but not in Shared Code 2.0.

---

# CoachSetDialogFilter

---

## Purpose

Defines a filter for a dialog box. The filter definition remains in memory until removed by CoachRemoveDialogFilter or until the macro ends. When the dialog box is initialized, macro execution moves to the dialog filter callback. (Not available in Shared Code 2.0.)

---

## Syntax

CoachSetDialogFilter(*DialogRegionName*; *Label*; *DefaultAction*)  
Example: CoachSetDialogFilter("WordPerfect.Tabs"; Tabsetcb@)

---

## Parameters

|                                             |                                                                                                                                                                                                                                                                |        |                  |            |                         |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------------------|------------|-------------------------|
| <i>DialogRegionName</i>                     | Specifies a dialog to filter.                                                                                                                                                                                                                                  |        |                  |            |                         |
| <i>Label</i>                                | Specifies the name of a callback label to go to when the dialog box is displayed.                                                                                                                                                                              |        |                  |            |                         |
| <i>DefaultAction</i><br>( <i>optional</i> ) | Specifies whether the default allows an event or throws it away. If a default action is not specified, all events are allowed.<br><table><tr><td>Allow!</td><td>Allow the event.</td></tr><tr><td>ThrowAway!</td><td>Do not allow the event.</td></tr></table> | Allow! | Allow the event. | ThrowAway! | Do not allow the event. |
| Allow!                                      | Allow the event.                                                                                                                                                                                                                                               |        |                  |            |                         |
| ThrowAway!                                  | Do not allow the event.                                                                                                                                                                                                                                        |        |                  |            |                         |

---

## Returns

When the dialog is invoked or a control on the dialog is activated, a variable array is created indicating the actions the user is taking. The dialog callback procedure can test each item in the array and perform appropriate operations.



The variable array's name is the same as the callback label. For example, the variable array of TabSetCB@ is TabSetCB[array item].

The variable array contains the following information depending on the message sent in Callback[5]:

| <b>Element</b> | <b>Description</b>                             |
|----------------|------------------------------------------------|
| Callback[1]    | Callback type (always 3 for a dialog callback) |
| Callback[2]    | Dialog name                                    |
| Callback[4]    | Dialog window handle                           |
| Callback[5]    | Message sent to the callback                   |

The following messages are valid and require the additional information shown.

272 = WM\_INITDIALOG. Sent when the dialog is first displayed. This message is for notification only and cannot be altered or prevented by the callback. If Callback[5] is 272, the following information is contained in the callback array:

Callback[6] = Window handle of the control that will receive the focus.

2 = WM\_DESTROY. Sent when the dialog goes down. This message is for notification only and cannot be altered or prevented by the callback. No additional information is contained in the callback array.

6 = WM\_ACTIVATE. Sent when the dialog is activated or inactivated. This message is for notification only and cannot be altered or prevented by the callback. If Callback[5] is 6, the following information is contained in the callback array:

Callback[6] = Activation state of the dialog

0 = WA\_INACTIVE (dialog becomes inactive)

1 = WA\_ACTIVE (dialog becomes active)

2 = WA\_CLICKACTIVE (dialog becomes active from a mouse click in the window)

Callback[8] = Minimized flag (A nonzero value means that the dialog is minimized.)

Callback[9] = Window handle of the window being activated or deactivated

3 = WM\_MOVE. Sent when the dialog is moved to a new location on screen. This message is for notification only and cannot be altered or prevented by the callback. If Callback[5] is 3, the following information is contained in the callback array:

Callback[8] = New X position of the window

Callback[9] = New Y position of the window

273 = WM\_COMMAND. Sent when the user interacts with a control or menu item owned by the dialog. WM\_COMMAND events can be altered by the callback. To prevent the dialog from acting on the control or menu item, set the return variable to 1. To enable the dialog to act on the menu item, set the return variable to 0. The default is 0. The information in the callback array depends on the menu or control type. If Callback[5] is 273, the following information is contained in the array:

Callback[3] = Control name (blank if the notification is from a menu item).

Callback[6] = Control or menu item ID.

Callback[9] = Control window handle (blank if the notification is from a menu item).

#### Menu Items

No additional information in the array.

#### ListBoxes

Callback[8] = LBN\_DBLCLK (2). List box item was double-clicked.

#### ComboBoxes

Callback[8] = CBN\_DBLCLK (2). Combo box item was double-clicked.

#### Button

Callback[8] = BN\_CLICKED (0). Button was clicked.

Callback[8] = BN\_DOUBLECLICKED (5). Button was double-clicked.

#### FileNameEntry

Callback[8] = FNEN\_DLGDDismiss (0). Open dialog was dismissed.

Callback[8] = FNEN\_DLGRAISE (1). Open dialog was displayed.

#### HotSpot

Callback[8] = BMPN\_CLICKED (0). Bitmap was clicked.

Callback[8] = BMPN\_DOUBLECLICKED (1). Bitmap was double-clicked.

---

## CoachRemoveDialogFilter

---

**Purpose** Removes a dialog box filter from memory. (Not available in Shared Code 2.0.)

---

**Syntax** CoachRemoveDialogFilter(*Dialog Name*)  
Example: CoachSetDialogFilter("WordPerfect.Tabs")

---

### Parameters

*Dialog Name* Specifies a dialog filter to remove.

---

## CoachAnimate

---

**Purpose** Specifies mouse actions or key strokes to execute without user intervention. This allows a Coach macro to demonstrate how to perform a function. You cannot use CoachAnimate when a filter is enabled.

---

**Syntax** CoachAnimate(*Type*; *Specification*)

---

**Parameters**

*Type* Specifies whether CoachAnimate animates the mouse or the keyboard. The enumerations, numeric equivalents, and descriptions are:

|           |   |                               |
|-----------|---|-------------------------------|
| Mouse!    | 1 | Mouse movements are animated. |
| Keyboard! | 2 | Keystrokes are animated.      |

*Specification* Wpstring: specifies actions to animate. It contains a keystring or a mouse string, depending on the value of *Type*. See *Keystring* and *Mouse String*.

---

## CoachFilterAdd

---

**Purpose**

Defines a filter. CoachFilterEnable enables defined filters.

Filters monitor: 1) mouse clicks in a specified area of the screen, 2) a single keystroke, 3) a combination of keystrokes (keystring), and 4) commands (tokens).

**Compound Filters**

To monitor more than one type of event, create a compound filter by using CoachFilterAdd several times, each time with the same filter name but with different filter specifications. A compound filter can monitor single keystrokes and mouse events, but cannot also monitor keystrings or commands. A compound filter can monitor several different commands, but cannot also monitor mouse events or keystrokes. A filter that monitors keystrings cannot be a compound filter.

For example, to create a compound filter that waits for either 1) a left mouse click in the OK button of a specified dialog, or 2) an Enter keystroke, use CoachFilterAdd twice with the same name each time. Specify a mouse string for the first command and a keystroke for the second.

You can enable one interface filter and one command filter at the same time (see *CoachEnableFilter*).

**Filter Callback Procedure**

Each filter can call at least one associated callback procedure which may give additional help when the user makes a mistake. A callback procedure begins with a label name and ends with RETURN.

After an enabled filter compares the user's actions with those defined in the filter, it determines whether to execute the filter callback procedure. *LabelCondition* in CoachFilterAdd determines when to call the callback: 1) the user's actions match the filter definition, 2) the user's actions do not match the filter definition, and 3) all cases.

When the macro executes a callback, return and array variables are created by the macro system and are accessible to the callback.

### Filter Array Variable

Contains information about the user's actions while a filter is enabled. The callback can access the array values for information about the event. The name of the array is formed from the label name. For example, if the label is LABEL1@, the array elements are LABEL1[1], LABEL1[2] and so on.

The information in the array depends on the type of callback receiving the array. There are three callback types: 1) mouse, 2) key (key and keystring), and 3) command.

### Mouse Filter Callbacks

| Element  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Array[1] | CBTYPE_MOUSE (0). Mouse callback type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Array[2] | Result of comparison between user's action and defined action. The possible values are:<br>-1 Error. The macro system did not complete the array.<br>0 Match. The user's actions exactly match the defined actions.<br>1 No match. The user's actions do not match the defined actions.<br>2 Partial Match. The user's actions partially match the defined actions. Possible only for a keyboard callback.<br>3 Regression. The event undoes previous events (for example, the Backspace key is pressed). A regression is not a match. A callback receives this value only if LabelType is All! or NonMatch!.<br>4 Quit Filter. The callback is not called. |
| Array[3] | Handle (hWnd) of the window in which the event occurs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Array[4] | Horizontal coordinate of the mouse position.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Array[5] | Vertical coordinate of the mouse position.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Array[6] | Windows flags for the message. Specific to the event message. (See the <i>Microsoft Windows Software Development Kit</i> .)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Array[7] | Windows event message, such as WM_LBUTTONDOWN or WM_LBUTTONUP.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

### Key Filter Callbacks

| Element  | Description                                                                                                                       |
|----------|-----------------------------------------------------------------------------------------------------------------------------------|
| Array[1] | CBTYPE_KEY (1). Key type.                                                                                                         |
| Array[2] | Filter Compare result (see <i>Mouse Filter Callback</i> above).                                                                   |
| Array[3] | Key code.                                                                                                                         |
| Array[4] | Windows flags for the event message. Specific to the event message. (See the <i>Microsoft Windows Software Development Kit</i> .) |

| <b>Element</b> | <b>Description</b>                                     |
|----------------|--------------------------------------------------------|
| Array[5]       | Windows event message, such as WM_KEYDOWN or WM_KEYUP. |

### **Command Callbacks**

| <b>Element</b> | <b>Description</b>                                                                              |
|----------------|-------------------------------------------------------------------------------------------------|
| Array[1]       | CBTYPE_TOKEN (2). Command type.                                                                 |
| Array[2]       | Filter compare result (see <i>Mouse Filter Callback</i> above).                                 |
| Array[3]       | Application that sent the command.                                                              |
| Array[4]       | Command MacroID. If a macro sent the command, this contains the macro ID.                       |
| Array[5]       | Unique command ID.                                                                              |
| Array[6]       | Number of parameters for this command.                                                          |
| Array[7]       | Command Flags. MAC_REQUIRED, MAC_REPEATING, and/or MAC_NODATA (see <i>Command Flags</i> below). |
| Array[8]       | Value of first parameter.                                                                       |
| Array[9]       | Value of second parameter.                                                                      |

...

### **Command Flags**

|               |        |                                         |
|---------------|--------|-----------------------------------------|
| MAC_REQUIRED  | 0x0100 | Required parameter.                     |
| MAC_REPEATING | 0x0400 | Parameter is part of a repeating group. |
| MAC_NODATA    | 0x8000 | No parameters.                          |

### **Filter Return Variable**

Specifies whether to ignore an event or send it to the application. Return variables have the same name as the callback label.

When a macro executes a callback's RETURN, the macro system checks the return variable, then passes the event to the application or throws it away. If the variable is undefined the macro system performs the default action:

| <b>Callback Type</b> | <b>Default Action</b>              |
|----------------------|------------------------------------|
| Mouse                | Throw away the event.              |
| Key                  | Throw away the event.              |
| Command              | Pass the event to the application. |

If the return variable is defined, the callback overrides the default action:

| <b>Value</b> | <b>Action</b>                      |
|--------------|------------------------------------|
| 0            | Pass the event to the application. |
| 1            | Throw away the event.              |

## Compound Filters and Callbacks

A compound filter may have several associated callback procedures, such as a different callback procedure for each filter specification (for example, one callback to handle mouse clicks and a second callback to handle key presses). An enabled compound filter checks its specifications in the order defined in the macro. After the filter determines that a callback procedure should be called, the remaining filter specifications are not checked.

## Performance

While the macro system processes a filter callback, all mouse movements, mouse clicks, and keystrokes are disabled until the callback returns.

---

## Syntax

CoachFilterAdd(Name; Label; LabelCondition; Type; Filter; AppName; Command; RepeatingGroup)

---

## Parameters

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                   |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|---|-------------------------------------|--------|---|-----------------------------------------------------------|------------|---|-------------------------------------------------------------------|----------|---|---------------------------|
| <i>Name</i>              | Wpstring: the name of the filter to define. If a filter with this name already exists, the new filter definition is appended to the existing definition to create a compound filter (see <i>Compound Filters</i> above).                                                                                                                                                                                                                                                                                                                                                         |                                                                   |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
|                          | Mouse! and Key! filters can be combined. KeyString! filters cannot be added to other filters. Command! filters can be appended to other Command! filters.                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                   |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| <i>Label</i>             | Wpstring: the macro label which is the name of a callback. Macro execution moves to this label as determined by <i>LabelCondition</i> (see <i>Filter Callback Procedure</i> above).                                                                                                                                                                                                                                                                                                                                                                                              |                                                                   |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| <i>LabelCondition</i>    | Specifies when to call the callback procedure marked by the <i>Label</i> label. The enumerations, numeric equivalents, and descriptions are:<br><table><tr><td>All!</td><td>1</td><td>All user actions call the callback.</td></tr><tr><td>Match!</td><td>2</td><td>All user actions that match the filter call the callback.</td></tr><tr><td>Nonmatch!</td><td>3</td><td>Only user actions that do not match the filter call the callback.</td></tr></table>                                                                                                                   | All!                                                              | 1 | All user actions call the callback. | Match! | 2 | All user actions that match the filter call the callback. | Nonmatch!  | 3 | Only user actions that do not match the filter call the callback. |          |   |                           |
| All!                     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | All user actions call the callback.                               |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| Match!                   | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | All user actions that match the filter call the callback.         |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| Nonmatch!                | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Only user actions that do not match the filter call the callback. |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| <i>Type</i>              | Specifies the type of user action to filter (mouse movements, keystrokes, or generated product commands) and what values to assign to some of the remaining parameters. The enumerations, numeric equivalents, and descriptions are:<br><table><tr><td>Mouse!</td><td>1</td><td>Filter mouse actions.</td></tr><tr><td>Key!</td><td>2</td><td>Filter single keystrokes.</td></tr><tr><td>KeyString!</td><td>3</td><td>Filter a series of keystrokes (keystring).</td></tr><tr><td>Command!</td><td>4</td><td>Filter a product command.</td></tr></table>                         | Mouse!                                                            | 1 | Filter mouse actions.               | Key!   | 2 | Filter single keystrokes.                                 | KeyString! | 3 | Filter a series of keystrokes (keystring).                        | Command! | 4 | Filter a product command. |
| Mouse!                   | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Filter mouse actions.                                             |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| Key!                     | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Filter single keystrokes.                                         |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| KeyString!               | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Filter a series of keystrokes (keystring).                        |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| Command!                 | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Filter a product command.                                         |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |
| <i>Filter (optional)</i> | Wpstring: specifies the actions to filter. For example, a mouse filter can specify that the user should click in the OK button of a particular dialog. A keystring filter can specify that the user should press Ctrl+B.<br><br>If <i>Type</i> is Mouse!, this parameter is a mouse string (see <i>Mouse String</i> below). If <i>Type</i> is Key!, this parameter is a single keystroke specification (see <i>Keystroke</i> below). If <i>Type</i> is KeyString!, this parameter is a keystring (see <i>Keystring</i> ). If <i>Type</i> is Command!, this parameter is ignored. |                                                                   |   |                                     |        |   |                                                           |            |   |                                                                   |          |   |                           |

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|---|---------------------------------------------|-----------|---|-------------------------------------------------|-------|---|----------------------------------------------|----------|---|-------------------------------------------------|-------------|---|----------------------------------------------------------|--------|---|-----------------------------------------------|-----------|---|----------------------------------|-----------|---|---------------------------------------------------------------------------|
| <i>AppName (optional)</i>        | Anstring: non-translatable application name, such as “WordPerfect”.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| <i>Command (optional)</i>        | Unsigned word: name of a product command to filter. Required if <i>Type</i> is Command!.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| <i>RepeatingGroup (optional)</i> | Filters parameter information for the product command specified in <i>Command</i> . Required only if <i>Type</i> is Command!.<br><br>If any parameter in a repeating group is assigned a value, all parameters of the repeating group must be assigned a value. You can specify more than one repeating group in the same CoachFilterAdd command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| <i>Parameter</i>                 | First parameter of repeating group. Unsigned word: The parameter number to filter. The Value command evaluates a parameter to a number. For example, the <i>Filename</i> parameter of FileOpenDlg is “Value(FileOpenDlg.Filename)”.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| <i>Compare</i>                   | Second parameter of repeating group. Specifies how to compare <i>Parameter</i> and <i>Value</i> . The enumerations, numeric equivalents, and descriptions are: <table> <tr> <td>Equal!</td> <td>1</td> <td><i>Parameter</i> is equal to <i>Value</i>.</td> </tr> <tr> <td>NotEqual!</td> <td>2</td> <td><i>Parameter</i> is not equal to <i>Value</i>.</td> </tr> <tr> <td>Less!</td> <td>3</td> <td><i>Parameter</i> is less than <i>Value</i>.</td> </tr> <tr> <td>Greater!</td> <td>4</td> <td><i>Parameter</i> is greater than <i>Value</i>.</td> </tr> <tr> <td>NotGreater!</td> <td>5</td> <td><i>Parameter</i> is less than or equal to <i>Value</i>.</td> </tr> <tr> <td>Exist!</td> <td>6</td> <td><i>Parameter</i> exists but can be any value.</td> </tr> <tr> <td>NotExist!</td> <td>7</td> <td><i>Parameter</i> does not exist.</td> </tr> <tr> <td>DontCare!</td> <td>8</td> <td><i>Parameter</i> may or may not exist. If it exists, it can be any value.</td> </tr> </table> | Equal!                                                                    | 1 | <i>Parameter</i> is equal to <i>Value</i> . | NotEqual! | 2 | <i>Parameter</i> is not equal to <i>Value</i> . | Less! | 3 | <i>Parameter</i> is less than <i>Value</i> . | Greater! | 4 | <i>Parameter</i> is greater than <i>Value</i> . | NotGreater! | 5 | <i>Parameter</i> is less than or equal to <i>Value</i> . | Exist! | 6 | <i>Parameter</i> exists but can be any value. | NotExist! | 7 | <i>Parameter</i> does not exist. | DontCare! | 8 | <i>Parameter</i> may or may not exist. If it exists, it can be any value. |
| Equal!                           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> is equal to <i>Value</i> .                               |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| NotEqual!                        | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> is not equal to <i>Value</i> .                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| Less!                            | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> is less than <i>Value</i> .                              |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| Greater!                         | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> is greater than <i>Value</i> .                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| NotGreater!                      | 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> is less than or equal to <i>Value</i> .                  |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| Exist!                           | 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> exists but can be any value.                             |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| NotExist!                        | 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> does not exist.                                          |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| DontCare!                        | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | <i>Parameter</i> may or may not exist. If it exists, it can be any value. |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |
| <i>Value</i>                     | Third parameter of repeating group. Contains any value type to compare with <i>Parameter</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                           |   |                                             |           |   |                                                 |       |   |                                              |          |   |                                                 |             |   |                                                          |        |   |                                               |           |   |                                  |           |   |                                                                           |

---

## CoachFilterDestroy

Removes a filter definition.

---

**Syntax** CoachFilterDestroy(*Name*)

**Parameters**

*Name* Wpstring: the name of a filter to destroy.

---

## CoachFilterEnable

Enables (activates) specified filters. Only one interface filter and one command filter can be enabled at a time.

When a filter is enabled, it compares a mouse movement, keystroke, or token to actions defined in the filter, then calls a callback procedure or passes the event unchanged.

When several filters are enabled, the filters are chained together. When an event occurs, the first filter filters the event. If that filter does not call a callback, the second filter filters the event, and so on. As soon as a filter calls a callback, no other filters can filter the event.

A filter must be defined by `CoachFilterAdd` before it can be enabled. An enabled filter remains enabled until it is disabled by `CoachFilterDisable` or destroyed by `CoachFilterDestroy`.

---

**Syntax**

`CoachFilterEnable(Command; Interface)`

---

**Parameters**

*Command (optional)*

Wpstring: the name of a command filter to enable.

*Interface (optional)*

Wpstring: the name of an interface filter to enable. Only one interface filter can be enabled at a time.

---

## CoachFilterDisable

---

**Purpose**

Disables an active filter. A filter definition remains in memory until `CoachFilterDestroy` destroys it, or until the macro ends.

---

**Syntax**

`CoachFilterDisable(Command; Interface)`

---

**Parameters**

*Command (optional)*

Wpstring: the name of a command filter to disable.

*Interface (optional)*

Wpstring: the name of an interface filter to disable.

---

## CoachGetRegionInfo

---

**Purpose**

Returns information about a named region, such as a window handle, rectangle, existence of the window, and control ID.



---

**Syntax**

CoachGetRegionInfo("Region"; Existvar:*exist*; Hwndvar:*hwnd*; Leftvar:*left*; Topvar:*top*; RightVar:*right*; Bottomvar:*bottom*)

---

**Parameters**

|                  |                                                                             |
|------------------|-----------------------------------------------------------------------------|
| <i>Region</i>    | Name of the region to test. For example:<br>"WordPerfect.Bookmark.GoToBttm" |
| <i>Existvar</i>  | Returns 0 if the region does not exist, and 1 if it does exist.             |
| <i>Hwndvar</i>   | Windows handle value of a specified region.                                 |
| <i>Leftvar</i>   | Screen coordinates for the left edge of a specified region.                 |
| <i>Topvar</i>    | Screen coordinates for the top edge of a specified region.                  |
| <i>Rightvar</i>  | Screen coordinates for the right edge of a specified region.                |
| <i>Bottomvar</i> | Screen coordinates for the bottom edge of a specified region.               |

---

**CoachMessageBox****Purpose**

Displays a message box. Message boxes display available options, such as a list of Coaches. Message boxes can include title, text, two bitmap images, radio buttons, check boxes, and pushbuttons.

Users can display more than one message box at a time and drag them to different positions on the screen.

**Message Box Callback Procedure**

Message boxes can have associated callback procedures. The callback procedure responds to events such as radio button, check box, and push button. For example, if the user clicks a message box Close button, the callback procedure calls CoachMessageBoxClose to dismiss the message box.

When macro execution moves to a callback, the macro system creates an array variable that the callback can access. The array name is formed from the label name. For example, the array elements of LABEL1@ are LABEL1[1], LABEL1[2], and so on. The array variable contains information about the message box and identifies the selected control.

**Element Description**

|          |                                                                                   |
|----------|-----------------------------------------------------------------------------------|
| Array[1] | CBTYPE_MESSAGEBOX (4). Message Box callback.                                      |
| Array[2] | Name of the message box specified in <i>MessageBoxName</i> of CoachMessageBox.    |
| Array[3] | Name of the control specified in <i>ControlName</i> of CoachMessageBox.           |
| Array[4] | A value that specifies whether a control is checked/pressed (1) or unchecked (0). |

---

## Syntax

CoachMessageBox(MessageBoxName; Title; Text; Library; IdBitmap1; IdBitmap2; Bmp2RedMask; Bmp2GreenMask; Bmp2BlueMask; BorderType; Bitmap1Position; Bitmap2Position; Top; Left; TitleFont; TitleFontSize; TextFont; TextFontSize; Modality; Label; ResultVariable; RepeatingGroup)

---

## Parameters

### *MessageBoxName*

Anstring: unique name of a message box. Passed to CoachMessageBoxClose to close the message box. CoachMessageBoxClose is required only if the message box modality is Modal! or Modeless! (see *Modality* below).

The same name can be used for more than one message. All messages with the same name close at the same time.

### *Title (optional)*

Wpstring: message box title. The title is displayed at the top of the message box, below the bitmap specified in *Bitmap1*. *TitleFont* and to specify font and font size.

### *Text (optional)*

Wpstring: formatted text:

| Character | Result              |
|-----------|---------------------|
| \n        | Insert hard return. |
| \b        | Toggle bold.        |
| \u        | Toggle underline.   |
| \i        | Toggle italic.      |

*Bitmap2Position* specifies text position. *TextFont* and *TextFontSize* specify font and font size.

### *Library (optional)*

Dword: the Windows handle of the DLL that contains the bitmaps specified in *IdBitmap1* and *IdBitmap2*. DLLOAD retrieves the handle. DLLFREE removes the DLL from memory.

### *IdBitmap1 (optional)*

Word: specifies a bitmap image, such as the Coach graphic, to display at the top of the message box.

Each bitmap specified in *IdBitmap1* must be followed by a monochrome version of the bitmap (transparency bitmap) in the DLL library. White areas in a transparency bitmap are transparent, black areas are not.

### *IdBitmap2 (optional)*

Word: specifies a bitmap image to display in the message box. *Bitmap2Position* specifies bitmap location.

Bitmaps specified by *IdBitmap2* do not require a transparency bitmap. *Bmp2RedMask*, *Bmp2GreenMask*, and *Bmp2BlueMask* specify a transparency color for *IdBitmap2* bitmaps to conserve disk space. Areas filled with the transparency color are transparent.

### *Bmp2RedMask (optional)*

Byte: *Bmp2RedMask*, *Bmp2GreenMask* and *Bmp2BlueMask* set red, green, and blue values (RGB) for the Bitmap2 transparency color.

The transparency color is the fill color that is transparent when the bitmap is displayed.

*Bmp2GreenMask*  
(optional)

See *Bmp2RedMask* above.

*Bmp2BlueMask*  
(optional)

See *Bmp2RedMask* above.

*Bitmap1Position*  
(optional)

Bitmap1 position in the message box. The enumerations, numeric equivalents, and descriptions are:

- Left! 1 Display bitmap in the top left corner. The top of the message box is flat.
- Top! 2 Display bitmap in a semicircle at the top.

*Bitmap2Position*  
(optional)

Bitmap2 position in the message box. The enumerations, numeric equivalents, and descriptions are:

- Left! 1 Align the left edge of the bitmap with the left edge of the message box. If the message box is longer than the bitmap, position the bitmap at the top of the message box. Text and controls will wrap to the right.
- Top! 2 Align the top edge of the bitmap directly under the message box title. If the message box is wider than the bitmap, display the bitmap at the left of the message box. Text and controls appear below the bitmap.
- Right! 3 Align the right edge of the bitmap with the right edge of the message box. If the message box is longer than the bitmap, display the bitmap at the top of the message box. Text and controls wrap to the left.
- Bottom! 4 Position the bottom edge of the bitmap above button controls, and below text and all other controls. Pushbuttons always appear at the bottom of the box. If the message box is wider than the bitmap, display the bitmap at the left of the message box.
- LeftCenter! 5 Center bitmap vertically.
- TopCenter! 6 Center bitmap horizontally.
- RightCenter! 7 Center bitmap vertically.
- BottomCenter! 8 Center bitmap horizontally.

*Top* (optional)

Word: specifies the percentage of the document window's vertical space above the center of the message box.

*Left* (optional)

Word: specifies the percentage of the document window's horizontal space left of the center of the message box.

*TitleFont* (optional)

Ansistring: the name of the Windows font to use to display the title. If no title font is specified, the default dialog font is used.

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>TitleFontSize</i><br>(optional)  | Word: the point size of the message box title.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| <i>TextFont</i> (optional)          | Anstring: the name of the Windows font to display text. If no font is specified, the default font (MS Sans Serif 8pt) is used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| <i>TextFontSize</i> (optional)      | Word: the point size of the message box text.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| <i>Modality</i>                     | The modality of the message box: 1) whether to dismiss the message box when a control is selected, 2) whether to call a callback procedure while a message box is displayed, and 3) whether the user can perform actions in the document window while the message box is displayed. The enumerations, numeric equivalents, and descriptions are: <table border="0" style="margin-left: 2em;"> <tr> <td style="vertical-align: top;">Simple!</td> <td style="vertical-align: top;">1</td> <td>Dismiss the message box when the user clicks a control. No callback procedure. User cannot perform actions in the document window while the message box is displayed.</td> </tr> <tr> <td style="vertical-align: top;">Modal!</td> <td style="vertical-align: top;">2</td> <td>Do not dismiss the message box until <code>CoachMessageBoxClose</code> is called. The callback procedure specified in <i>Label</i> is called when the user selects a control. The user cannot perform actions in the document window while the message box is displayed.</td> </tr> <tr> <td style="vertical-align: top;">Modeless!</td> <td style="vertical-align: top;">3</td> <td>Do not dismiss the message box until <code>CoachMessageBoxClose</code> is called. The callback procedure specified in <i>Label</i> is called when the user selects a control. The user can perform actions in the document window while the message box is displayed.</td> </tr> </table> | Simple!                                                                                                                                                                                                                                                                  | 1 | Dismiss the message box when the user clicks a control. No callback procedure. User cannot perform actions in the document window while the message box is displayed. | Modal! | 2 | Do not dismiss the message box until <code>CoachMessageBoxClose</code> is called. The callback procedure specified in <i>Label</i> is called when the user selects a control. The user cannot perform actions in the document window while the message box is displayed. | Modeless! | 3 | Do not dismiss the message box until <code>CoachMessageBoxClose</code> is called. The callback procedure specified in <i>Label</i> is called when the user selects a control. The user can perform actions in the document window while the message box is displayed. |
| Simple!                             | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Dismiss the message box when the user clicks a control. No callback procedure. User cannot perform actions in the document window while the message box is displayed.                                                                                                    |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| Modal!                              | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Do not dismiss the message box until <code>CoachMessageBoxClose</code> is called. The callback procedure specified in <i>Label</i> is called when the user selects a control. The user cannot perform actions in the document window while the message box is displayed. |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| Modeless!                           | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Do not dismiss the message box until <code>CoachMessageBoxClose</code> is called. The callback procedure specified in <i>Label</i> is called when the user selects a control. The user can perform actions in the document window while the message box is displayed.    |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| <i>Label</i> (optional)             | <p>Wpstring: the label name of a message box callback procedure. Set this parameter only if the message box modality is Modal! or Modeless!. If <i>Modality</i> is Simple!, this parameter is ignored.</p> <p>The message box calls this label when the user selects a control (radio button, check box, or pushbutton) on the message box. Also see <i>Message Box Callback Procedure</i> above.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| <i>ResultVariable</i><br>(optional) | Variable: used only if <i>Modality</i> is set to Simple!. When a message box's modality is simple, the message box is automatically dismissed when the user selects a control. The result variable specified in this parameter holds the name of the control ( <code>ControlName</code> ) that was selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |
| <i>RepeatingGroup</i><br>(optional) | <p>Specifies the controls (such as radio buttons, check boxes, and pushbuttons) to display in the message box. Each repeating group contains three parameters which hold the information for one control. You can specify multiple controls.</p> <p>Controls always appear beneath the message box text (see <i>Text</i>). Regardless of the order in which the controls are specified, they appear in the message box in this order: radio buttons, check boxes, and pushbuttons. If <code>Bitmap2Position</code> is Bottom!, <code>bitmap2</code> displays below radio buttons and check boxes, and above pushbuttons (pushbuttons always display at the bottom of the message box).</p> <p>If one parameter of a repeating group is specified, all three must be specified.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                          |   |                                                                                                                                                                       |        |   |                                                                                                                                                                                                                                                                          |           |   |                                                                                                                                                                                                                                                                       |

|                    |                                                                                                                                                                                                                                                                                                       |              |   |           |   |         |   |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|---|-----------|---|---------|---|
| <i>ControlName</i> | First Parameter of Repeating Group. Ansistring: a control name. The name does not appear on the message box. If this control is selected in the message box, the control name is displayed in Array[3]. If the message box modality is Simple!, the control name is copied to <i>ResultVariable</i> . |              |   |           |   |         |   |
| <i>ControlType</i> | Second Parameter of Repeating Group. Indicates a control type. The enumerations, numeric equivalents, and descriptions are: <table> <tr> <td>RadioButton!</td> <td>1</td> </tr> <tr> <td>CheckBox!</td> <td>2</td> </tr> <tr> <td>Button!</td> <td>3</td> </tr> </table>                              | RadioButton! | 1 | CheckBox! | 2 | Button! | 3 |
| RadioButton!       | 1                                                                                                                                                                                                                                                                                                     |              |   |           |   |         |   |
| CheckBox!          | 2                                                                                                                                                                                                                                                                                                     |              |   |           |   |         |   |
| Button!            | 3                                                                                                                                                                                                                                                                                                     |              |   |           |   |         |   |
| <i>ControlText</i> | Third Parameter of Repeating Group. Ansistring: control text. Text is displayed to the right of radio buttons and check boxes, and inside push buttons.                                                                                                                                               |              |   |           |   |         |   |

---

## CoachMessageBoxClose

**Purpose** Closes a message box. Required only for Modeless! or Modal! message boxes (see *Modality* in CoachMessageBox).

**Syntax** CoachMessageBoxClose(*MessageBoxName*)

**Parameters**

*MessageBoxName* Ansistring: name of a message box to close. This should be the same name as in *MessageBoxName* of CoachMessageBox.

---

## CoachPrompt

**Purpose** Displays a prompt. The prompt can display a message and a bitmap image that points to an item, such as a prompt that points to the File menu with the text “Click the File menu item.”

The prompt location is set relative to a region (see *Location* and *Region*), or set to a fixed position in the document window.

When you set a location relative to a region, you can have a prompt display to the right, left, upper right, upper left, lower right, or lower left of the region. The system adjusts the size of the prompt box to fit in the display area. If the prompt is too large, the system checks for a suitable location by rotating the prompt clockwise. For example, if the prompt is to display to the right of a region, but there isn't room, the system checks to the lower right. If it will not fit there, the system checks the lower left, and so on.

CoachPrompt points to items with bitmap images. The bitmap image is displayed between the prompt box and the item.

*IdBitmap* specifies the bitmap image ID. For example, if the prompt is displayed at the upper left, and the bitmap is a pointing hand, the fingers point down and to the right.

Bitmaps for all 6 prompt locations must be defined in order, starting with the upper left bitmap and proceeding clockwise. The transparency bitmap must immediately follow each bitmap. *IdBitmap* should always contain the first bitmap ID (upper left).

---

## Syntax

CoachPrompt(*PromptName*; *Message*; *Font*; *FontSize*; *Library*; *IdBitmap*; *Location*; *Top*; *Left*; *Region*)

---

## Parameters

*PromptName* Ansistring: a unique name for the prompt.

*Message (optional)* Wpstring: message to display in the prompt. Formatting codes are optional:

| Character | Result             |
|-----------|--------------------|
| \n        | Insert Hard Return |
| \b        | Toggle Bold        |
| \u        | Toggle Underline   |
| \i        | Toggle Italic      |

*Font (optional)* Ansistring: name of the Windows font for prompt text. If no font is specified, the default dialog font is used.

*FontSize (optional)* Word: point size of the prompt text. If no font is specified, the default font (MS Sans Serif 8pt) is used.

*Library (optional)* Dword: the Windows handle of the DLL that contains the bitmap to display outside the prompt. DLOAD retrieves the handle. DLLFREE removes the DLL from memory.

*IdBitmap (optional)* Word: the identifier of the first bitmap image defined for CoachPrompt.

In the bitmap DLL library each bitmap specified by *IdBitmap* must be immediately followed by a transparency bitmap. A transparency bitmap is a monochrome version of a bitmap. White areas in the transparency bitmap are transparent, black areas are not.

*Location (optional)* Specifies where to display the prompt in relation to the region (see *Region*). The enumerations, numeric equivalents, and descriptions are:

|             |   |                                                             |
|-------------|---|-------------------------------------------------------------|
| UpperLeft!  | 1 | Display prompt upper left.                                  |
| UpperRight! | 2 | Display prompt upper right.                                 |
| Right!      | 3 | Display prompt right.                                       |
| LowerRight! | 4 | Display prompt lower right.                                 |
| LowerLeft!  | 5 | Display prompt lower left.                                  |
| Left!       | 6 | Display prompt left.                                        |
| Specify!    | 7 | Display prompt as specified by <i>Top</i> and <i>Left</i> . |

*Top (optional)*

Word: specifies the percentage of the document window's vertical space above the center of the message box. Required only if *Location* is Specify!.

*Left (optional)*

Word: specifies the percentage of the document window's horizontal space left of the center of the message box. Required only if *Location* is Specify!.

*Region (optional)*

Wpstring: the named region the prompt points to. Not required if *Top* and *Left* are specified.

---

## CoachPromptClose

---

**Purpose** Closes a prompt.

---

**Syntax** CoachPromptClose(*PromptName*)

---

### Parameters

*PromptName* Ansistring: name of the prompt to close. This is the same name specified in *PromptName* of CoachPrompt.

---

## Keystring

Specifies one or more keystrokes in CoachAnimate and CoachFilterAdd. In CoachAnimate, a keystring specifies keys to press during animation. In CoachFilterAdd, a keystring specifies keystrokes to filter.

The following metasymbols specify a key or keystring in CoachFilterAdd and CoachAnimate.

{VKnnn} // nnn = ANSI character number

{Alt}

{Ctrl}

{Control}

{Shift}

{0} – {9} // Digits

{A} – {Z} // Alphabet

{F1} – {F16} // Function keys

{NumLock}

{NumAdd}

{NumSubtract}

{NumMultiply}

{NumDivide}

{NumDecimal}

{NumEnter}

```

{Num0} – {Num9} // Numpad numbers

{Left}
{Right}
{Up}
{Dn}
{Down}

{PgDn}
{PageDown}
{PgUp}
{PageUp}

{Bksp}
{Backspace}
{Break} // Cancel
{CapsLock}
{Clear}
{Del}
{Delete}
{End}
{Enter}
{Esc}
{Escape}
{Help} // VK Help key
{Home}
{Ins}
{Insert}
{Minus}
{Pause}
{ScrLock}
{ScrollLock}
{PrintScrn}
{PrintScreen}
{Space}
{Tab}

{LeftBrace} // "{"
{RightBrace} // "}"

{LeftButton} // Mouse clicks
{MiddleButton}
{RightButton}

```

Metasymbols are translated into keydown messages (WM\_KEYDOWN or WM\_SYSKEYDOWN) and keyup messages (WM\_KEYUP or WM\_SYSKEYUP).

Metasymbols are not case-sensitive. Spaces, plus (+) and minus (–) symbols are delimiters. For example, "A B C Enter" in CoachAnimate sends "ABC[Enter]" to the application. A filter that specifies this keystring waits for the user to press "ABC[Enter]".



Plus (+) and minus (–) operators combine metasymbols. The + delays the keyup message of the first keystroke until the second keystroke is processed. For example, {Shift + Tab} presses and holds Shift while Tab is pressed and released. The Shift key is then released.

If a metasymbol does not follow +, then keys previously combined with + remain down until released. For example, {Shift+Control+} leaves Shift and Control down.

The – releases keys. For example, {–Control–Shift} releases Control, then Shift.

The + and – can be combined. For example, {Shift+Del–Del+Ins} cuts selected text to the Clipboard (Shift+Del), then pastes it at the insertion point (Shift+Ins).

Example:

```
{Home Shift+End}{Control+B End}{Enter}{Alt+F O} *.doc {Enter}{Escape}{Ctrl+Home}{Shift+}{Ctrl+End
Del}{–Shift}
```

Result:

1. Move to the beginning of the line and select the entire line.
2. Bold the line and move to the end of the line (cancelling the selection).
3. Create a new line.
4. Call the File Open dialog, list all \*.DOC files, and then exit the dialog.
5. Go to the top of the document.
6. Hold down Shift.
7. Go to the end of the text and copy the selection to the Clipboard.
8. Release Shift.

---

## Mouse String

|                |                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Purpose</b> | Represents a mouse event or action.<br><br>CoachAnimate specifies the animated mouse actions. CoachFilterAdd specifies the filtered mouse actions.                                                                                                                                                                                                                                                             |
| <b>Syntax</b>  | The syntax for mouse events for CoachAnimate is:<br><br>[Ctrl Control] [Shift] [Alt]<br>[Left Middle Right]<br>Click DoubleClick Press Release Move<br>[To In Inside On] [NamedRegion]<br>[At (x,x) (x,x,x,x)]<br><br>"Outside," "OutOf," and "Off" are supported in CoachFilterAdd and "Move" is not supported.<br><br>[Ctrl Control] [Shift] [Alt]<br>[Left Middle Right]<br>Click DoubleClick Press Release |

[To|In|Inside|On|Outside|OutOf|Off] [NamedRegion]  
[At (x,x)|(x,x,x,x)]

Arguments are not case-sensitive. Brackets ([]) enclose optional values. A vertical bar | separates values when only one can be chosen. A space separates arguments.

---

## Arguments

Mouse string arguments:

### *Keys (optional)*

[Ctrl|Control] [Shift] [Alt] Specifies keys to press with the mouse event, such as holding down Shift and Control when the mouse is clicked. These values can occur in any order.

### *Button (optional)*

[Left|Middle|Right] Indicates whether to use the left, middle, or right button. Left is the default.

### *Action*

Click|DoubleClick|Press|Release|Move Indicates how to click or press the mouse button, or whether to move the mouse pointer.

| Argument | Meaning |
|----------|---------|
|----------|---------|

|       |            |
|-------|------------|
| Click | One click. |
|-------|------------|

|             |               |
|-------------|---------------|
| DoubleClick | Double-click. |
|-------------|---------------|

|       |                             |
|-------|-----------------------------|
| Press | Press and hold button down. |
|-------|-----------------------------|

|         |                                                                                                     |
|---------|-----------------------------------------------------------------------------------------------------|
| Release | Release button after it is pressed. Use Press with Release for <i>drag</i> (use two mouse strings). |
|---------|-----------------------------------------------------------------------------------------------------|

|      |                                                                 |
|------|-----------------------------------------------------------------|
| Move | Move pointer to specified location. Valid only in CoachAnimate. |
|------|-----------------------------------------------------------------|

### *Relative Location*

[To|In|Inside|On|Outside|OutOf|Off] Specifies a screen position for a mouse event.

| Argument | Meaning |
|----------|---------|
|----------|---------|

|    |                                                                                                   |
|----|---------------------------------------------------------------------------------------------------|
| To | If the Action argument is Move, To moves the pointer to the named region. Otherwise, To means In. |
|----|---------------------------------------------------------------------------------------------------|

|    |                                                                                          |
|----|------------------------------------------------------------------------------------------|
| In | Mouse action occurs within the specified named region (see <i>Named Region</i> , below). |
|----|------------------------------------------------------------------------------------------|

|        |             |
|--------|-------------|
| Inside | Same as In. |
|--------|-------------|

|    |             |
|----|-------------|
| On | Same as In. |
|----|-------------|

|         |                                                                                           |
|---------|-------------------------------------------------------------------------------------------|
| Outside | Mouse action occurs outside the specified named region. Supported only in CoachFilterAdd. |
|---------|-------------------------------------------------------------------------------------------|

|       |                                                    |
|-------|----------------------------------------------------|
| OutOf | Same as Outside. Supported only in CoachFilterAdd. |
|-------|----------------------------------------------------|

|     |                                                    |
|-----|----------------------------------------------------|
| Off | Same as Outside. Supported only in CoachFilterAdd. |
|-----|----------------------------------------------------|

### *Named Region*

[NamedRegion] Specifies a window or location where the mouse action occurs. Required only if a relative location is specified. For example, you may want to specify a mouse click in the Open button of the Open Dialog. The named region specifies a name for the Open button.

Named regions are defined by the application. The region consists of the application name, followed by a period (.), followed by additional words that narrow the named

region to the appropriate window. For example, the named region of the Open button on the Open dialog of WordPerfect is: WordPerfect.FileOpenDlg.Open.

Use the WordPerfect Help system to find a region name. First edit your BIF file to enable Help Information. The BIF editor, BIFED20.EXE, should be in the WPC20 directory.

- 1 Exit WordPerfect and start Program Manager.
- 2 Choose **File**, then **Run**.
- 3 In the BIF Edit dialog box, choose **File**, then **Open**.
- 4 Select the desired file.
- 5 Select **WP Shared Code**.
- 6 Choose **Insert** to add a new section.
- 7 Type **Help** in the **Section Name** entry box.
- 8 Type **Help Information** in the **Item** entry box. The Item Type is Boolean.
- 9 Select **True** for the **Value**.
- 10 Choose **File**, then **Save**.
- 11 Choose **File**, then **Exit**.
- 12 Start WordPerfect.

To find a named region:

- 1 Press Shift+F1.
- 2 Click the desired menu item or dialog box.

A Help Information screen is displayed. The Application and Keyword items display the named region.

Help Information for dialog boxes returns application and dialog box names, but not named regions for controls on the dialog. See the list of dialog box named regions in the Appendix for control named regions.

Buttons are named for the text on them, plus Bbtn. For example, the Help button in the HeadersFooters dialog box is WordPerfect.HeadersFooters.HelpBbtn. Named region names are case-sensitive.

[At (x,x)|(x,x,x,x)] Specifies an exact point (x,x), or rectangle (x,x,x,x) where a mouse event occurs. Point and rectangle positions are relative to the window specified in *Named Region*. Pixels are the default unit of measure. For example, in CoachAnimate,

Move To WordPerfect.FileOpenDlg.Open At (10,5)

moves the pointer to a point 10 pixels right and 5 pixels down from the upper left corner of the Open button.

*Narrowed Location*  
(optional)

You can specify the percentage from the upper left corner of the window by placing a percent sign % after values. For example,

```
Move To WordPerfect.FileOpenDlg.Open At (50%, 50%)
```

moves the pointer to the center of the Open button.

To specify dialog units, place the letter d after values.

### **Examples**

Three mouse string examples follow. Named regions do not represent actual WordPerfect names.

Example:

```
Shift Right DblClick
```

Result: Shift is held down while the right mouse button is double-clicked (the mouse pointer may be anywhere on the screen).

Example:

```
Move at (50%, 50%)
```

Result: This mouse string is invalid in CoachFilterAdd. When this string is specified in CoachAnimate, the mouse pointer moves to the center of the display.

Example:

```
Press on WordPerfect.Ruler.Tab1
Release on WordPerfect.EditWindow
```

Result: These mouse strings must be specified in two consecutive commands. The first tab setting on the ruler is dragged off the ruler (deleted).

# Appendixes

.....

## Appendixes

# Appendix A: Reserved Words

You should not use reserved words as names for variables, arrays, labels, functions, or procedures.

|                 |                |
|-----------------|----------------|
| ADDRESS         | DLLLOAD        |
| AND             | DWORD          |
| ANSISTRING      | ELSE           |
| APPACTIVATE     | ENDAPP         |
| APPEXECUTE      | ENDFOR         |
| APPEXECUTEEXT   | ENDFUNC        |
| APPOCCATE       | ENDIF          |
| APPLICATION     | ENDIFPLATFORM  |
| ASSERT          | ENDPROC        |
| ASSERTCANCEL    | ENDPROMPT      |
| ASSERTERROR     | ENDSWITCH      |
| ASSERTNOTFOUND  | ENDWHILE       |
| ASSIGN          | ERROR          |
| BEEP            | ERRORCONDITION |
| BOOL            | ERRORNUMBER    |
| BREAK           | ERROROFF       |
| CALL            | ERRORON        |
| CANCEL          | EXISTS         |
| CANCELCONDITION | FALSE          |
| CANCELOFF       | FILE           |
| CANCELON        | FOR            |
| CASE            | FOREACH        |
| CASEOF          | FORNEXT        |
| CENTIMETERS     | FRACTION       |
| CHAIN           | FUNCTION       |
| CHARLEN         | GETNUMBER      |
| CHARPOS         | GETSTRING      |
| CONTINUE        | GETUNITS       |
| CTON            | GLOBAL         |
| DDEEXECUTE      | GO             |
| DDEEXECUTEEXT   | HIWORD         |
| DDEINITIATE     | IF             |
| DDEPOKE         | IFPLATFORM     |
| DDEREQUEST      | INCHES         |
| DDETERMINATE    | INDIRECT       |
| DDETERMINATEALL | INPUT          |
| DECLARE         | INTEGER        |
| DEFAULT         | LABEL          |
| DEFAULTUNITS    | LENGTH         |
| DIGIT           | LETTER         |
| DISCARD         | LOCAL          |
| DIV             | LOWORD         |
| DLLCALL         | MENU           |
| DLLFREE         | MENULIST       |

MILLIMETERS  
MOD  
NEST  
NEXT  
NEWDEFAULT  
NOT  
NOTFOUND  
NOTFOUNDCONDITION  
NOTFOUNDOFF  
NOTFOUNDON  
NTOC  
NUMSTR  
OEMSTRING  
OFF  
ON  
ONCANCEL  
ONDDEADVISE  
ONERROR  
ONNOTFOUND  
OR  
PAUSE  
PERSIST  
PERSISTALL  
POINTS  
POSITION  
PROCEDURE  
PROMPT  
PROTOTYPE  
QUIT  
REAL  
REPEAT  
RETURN  
RETURNCANCEL  
RETURNERROR  
RETURNNOTFOUND  
RUN  
SENDKEYS  
SPEED  
STRING  
STRLEN  
STRNUM  
STRPOS  
STRUNIT  
SUBCHAR  
SUBSTR  
SWITCH  
TOWER  
TOUPPER  
TRUE  
UNITSTR  
UNTIL  
USE  
VALUE  
VARERRCHK  
VARERRCHKOFF  
VARERRCHKON  
VOID  
WAIT  
WHILE  
WIN  
WORD  
WPSTRING  
WPUNITS  
WP1200THS  
XOR



## Appendix B: Shipping Macros

The following list contains a brief description of the macros that are included with WordPerfect 6.0 for Windows. Most of the macros are represented by buttons on Button Bars. You can also play them separately.

### Macros on the Design Tools Button Bar

| Macro Name   | Description                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| CAPITAL.WCM  | Capitalizes the first letter of the current word.                                                                    |
| DROPCAP.WCM  | Changes the first letter in a paragraph to a drop cap.                                                               |
| ENDFOOT.WCM  | Converts endnotes to footnotes in the document or in selected text.                                                  |
| FILESTMP.WCM | Places the filename and path of the current document in a header or footer.                                          |
| FONTDN.WCM   | Decreases the font size of the selected text by 2 points.                                                            |
| FONTUP.WCM   | Increases the font size of the selected text by 2 points.                                                            |
| FOOTEND.WCM  | Converts footnotes to endnotes in the document or selected text.                                                     |
| PARABRK.WCM  | Inserts graphical paragraph breaks between paragraphs.                                                               |
| PGBORDER.WCM | Uses graphics (.WPG files) to create decorative page borders.                                                        |
| REVERSE.WCM  | Displays white text on black background (or other color options as set by the user) in selected text or table cells. |
| WATERMRK.WCM | Prompts user for text or watermark graphic to place in a watermark (centered vertically and horizontally).           |

### Macros on the Legal Button Bar

| Macro Name   | Description                                                                                                |
|--------------|------------------------------------------------------------------------------------------------------------|
| PLEADING.WCM | Creates pleading lines and numbers.                                                                        |
| WATERMRK.WCM | Prompts user for text or watermark graphic to place in a watermark (centered vertically and horizontally). |

### Macros on the Macro Tools Button Bar

| Macro Name  | Description                                                                              |
|-------------|------------------------------------------------------------------------------------------|
| MCVTWIN.WCM | Converts WordPerfect 5.1 and 5.2 for Windows macros to 6.0 format.                       |
| LINENUM.WCM | Places the insertion point at a specified line and character for troubleshooting macros. |

### Macro on the Page Button Bar

| Macro Name   | Description                                                                                   |
|--------------|-----------------------------------------------------------------------------------------------|
| PAGEXOFY.WCM | Places "Page x of y pages" (or other pagination formats) in a specified position on the page. |

---

**Macros on the Utilities  
Button Bar**

| <b>Macro Name</b> | <b>Description</b>                                                                                            |
|-------------------|---------------------------------------------------------------------------------------------------------------|
| ABBREV.WCM        | Lets you create and expand multiple abbreviations.                                                            |
| ADRS2MRG.WCM      | Copies the address book (for automated templates) into a merge data file.                                     |
| ADRSBOOK.WCM      | Opens the address book.                                                                                       |
| CLIPBRD.WCM       | Opens the Windows Clipboard.                                                                                  |
| CLOSEALL.WCM      | Closes all open documents and prompts you to save the modified ones.                                          |
| EPUB.WCM          | Publishes current document electronically.                                                                    |
| EXPNDALL.WCM      | Expands all abbreviations in the document in one step.                                                        |
| GOTODOS.WCM       | Opens a DOS window.                                                                                           |
| READCLP.WCM       | Reads text in the Clipboard. Requires ProVoice or Monologue text-to-speech software.                          |
| READFILE.WCM      | Reads a text file or plays a .WAV, .AVI or .MID file. Requires ProVoice or Monologue text-to-speech software. |
| READSEL.WCM       | Reads text selected in the document. Requires ProVoice or Monologue text to speech software.                  |
| SAVEALL.WCM       | Prompts you to save open documents.                                                                           |

---

**Other Macros not on a  
Button Bar**

| <b>Macros Name</b> | <b>Description</b>                                                                                                                            |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| ALLFONTS.WCM       | Creates a document listing every font available to the current printer and some sample text in each font.                                     |
| _AUTOFIL.WCM       | Used by ExpressDocs templates created in WPWIN 6.0 to automate the template fill-in process. Templates created in 6.0a do not use this macro. |
| COPYQPW.WCM        | Copies a Quattro Pro Notebook selection to WordPerfect.                                                                                       |
| CTRLM.WCM          | Opens the Macro Command Inserter.                                                                                                             |
| CTRLSFTE.WCM       | For Equation Editor - prompts for FROM and TO values.                                                                                         |
| PROMPTS.WCM        | Helps you create prompts for automated templates.                                                                                             |
| QCSETUP.WCM        | If QuickCorrect is not found on the Tools menu, this macro adds it.                                                                           |
| QUICKCPY.WCM       | Copies selected item from Quattro Pro to WordPerfect.                                                                                         |
| TCONVERT.WCM       | Converts WPWIN 6.0 templates that used the AUTOFILL.WCM macro.                                                                                |
| TRANSPOS.WCM       | Transposes the two characters preceding the insertion point.                                                                                  |

## Appendix C: Windows Only Commands

The following WordPerfect 6.0 for Windows commands do not have DOS equivalents.

|                      |                             |
|----------------------|-----------------------------|
| AbbreviationCreate   | CloseGraphicsControlBar     |
| AbbreviationDelete   | CloseNoSave                 |
| AbbreviationDlg      | CodePage                    |
| AbbreviationExpand   | CodePageDlg                 |
| AboutDlg             | ColumnBorderDlg             |
| AppMaximize          | CommentInformation          |
| AppMinimize          | CommentInsertTime           |
| AppMove              | CommentInsertUserInitials   |
| AppRestore           | CommentInsertUserName       |
| AppSize              | CreateMapFile               |
| AppTaskList          | CtrlBreak                   |
| AssociateDlg         | CustomBoxDlg                |
| BarcodeDlg           | DDEChangeUpdateMode         |
| BindingOptionsDlg    | DDECreateLink               |
| BorderFillColors     | DDECreateLinkDlg            |
| BoxBorder            | DDEDeleteLink               |
| BoxCaptionEditEnd    | DDEDeleteLinkDlg            |
| BoxCaptionFormatting | DDEEditLink                 |
| BoxEquationColor     | DDEEditLinkDlg              |
| BoxEquationFontSize  | DDEPasteLink                |
| BoxEquationRedisplay | DDEUpdateLink               |
| BoxFill              | DDEUpdateLinkDlg            |
| BoxInitialStyleDlg   | Display                     |
| BoxOLEEditObject     | DocCompareAddMarkingsDlg    |
| BoxPageOffset        | DocCompareRemoveMarkingsDlg |
| BoxPosDlg            | DocInitialFontDlg           |
| BoxTextEdit          | DocMaximize                 |
| BoxTextEditEnd       | DocMinimize                 |
| BoxUpdateDisplay     | DocMove                     |
| BoxWrapDlg           | DocNext                     |
| BulletDlg            | DocPrevious                 |
| ButtonBarCreateDlg   | DocRedlineMethodDlg         |
| ButtonBarEditDlg     | DocRestore                  |
| ButtonBarOptionsDlg  | DocSize                     |
| ButtonBarSetFont     | DocSummaryConfig            |
| ButtonBarSetLocation | DocSummarySaveAs            |
| ButtonBarSetRows     | DocSumSet                   |
| ButtonBarSetStyle    | DocSumSetFull               |
| ButtonBarShow        | DraftZoom100                |
| ButtonBarShowToggle  | DraftZoomMargin             |
| CaseToggle           | EditComment                 |
| CenterPageDlg        | EditCopy                    |
| Char                 | EditCut                     |
| CharMappingDlg       | EditPaste                   |

|                           |                               |
|---------------------------|-------------------------------|
| EndnoteOptions            | GraphicCustomMenuDlg          |
| EquationBaseFontDlg       | GraphicEditBox                |
| EquationCreate            | GraphicsBoxCaptionDlg         |
| EquationPalette           | GraphicsBoxContentDlg         |
| EquationPaletteToggle     | GraphicsBoxFindDlg            |
| EquationRetrieveDlg       | GraphicsBoxSizeDlg            |
| EquationZoomDlg           | GraphicsFigureEditAllDlg      |
| ExpandDlg                 | GraphicsLineUpdateDisplay     |
| FeatureBarPopDown         | GraphicsStylesDlg             |
| FigureNoMouse             | HardPageBreakInsert           |
| FigurePaletteDlg          | HeaderFooterDistanceDlg       |
| FigurePaletteDlgEnd       | HeaderFooterDlg               |
| FigureReset               | HeaderFooterPlacement         |
| FigureRetrieveDlg         | HeaderFooterPlacementDlg      |
| FileChangeAttributes      | HelpCoachDlg                  |
| FileChangeDir             | HelpContents                  |
| FileInsert                | HelpContextSensitive          |
| FileInsertDlg             | HelpHowDoI                    |
| FileMove                  | HelpHowToUseHelp              |
| FileRemoveDirectory       | HelpKeystrokes                |
| FileSaveAsDlg             | HelpMacroManual               |
| FindDlg                   | HelpSearchForHelpOn           |
| FindFiles                 | HelpTutorial                  |
| FontDoubleUnderlineToggle | HelpWhatIs                    |
| FontExtraLargeToggle      | HideBars                      |
| FontFineToggle            | HLineCreate                   |
| FontHiddenToggle          | HypertextControlBar           |
| FontItalicToggle          | HypertextCreateDlg            |
| FontLargeToggle           | HypertextExecute              |
| FontOutlineToggle         | Hyphen                        |
| FontRedlineToggle         | ImportCreateDlg               |
| FontShadowToggle          | ImportDlg                     |
| FontSmallCapsToggle       | ImportDoEdit                  |
| FontSmallToggle           | ImportDoImport                |
| FontStrikeoutToggle       | ImportDoLink                  |
| FontSubscriptToggle       | ImportDoUpdate                |
| FontSuperscriptToggle     | ImportEditDlg                 |
| FontUnderlineToggle       | ImportOptionsDlg              |
| FontVeryLargeToggle       | ImportSetAsciiEncap           |
| FootnoteNewNumberDlg      | ImportSetAsciiFieldDelimiter  |
| FootnoteOptions           | ImportSetAsciiRecordDelimiter |
| ForcePageDlg              | ImportSetAsciiStrip           |
| GetAcceleratorText        | ImportSetDestination          |
| GetData                   | ImportSetDictPath             |
| GraphicBoxStyleDlg        | ImportSetFields               |
| GraphicCreateBox          | ImportSetFileName             |
| GraphicCreateChart        | ImportSetFilePath             |
| GraphicCreateTextArt      | ImportSetIncludeNames         |

|                        |                        |
|------------------------|------------------------|
| ImportSetPanel         | MarkIndexUpdateList    |
| ImportSetPassword      | MarkList               |
| ImportsetQuery         | MarkListUpdateList     |
| ImportSetRange         | MarkTableOfAuthorities |
| ImportSetServer        | MarkTableOfContents    |
| ImportSetSource        | MarkToAUpdateList      |
| ImportSetSQLQuery      | MasterDocSubdocDlg     |
| ImportSetUserID        | MatchLimit             |
| ImportUpdateDlg        | MenuAddItem            |
| IndexDefineDlg         | MenuDeleteItem         |
| InhibitInput           | MenuSelect             |
| InitialCodesStyleDlg   | MenuSelectDlg          |
| InsertDefaultBullet    | MergeDlg               |
| InsertOtherCodesDlg    | MergeEndField          |
| InvokeDialog           | MergeEndRecord         |
| JustifyAll             | MergeFilesDlg          |
| JustifyCenter          | MergeFormDlg           |
| JustifyFull            | MergeTableDataDlg      |
| JustifyLeft            | MergeTextDataDlg       |
| JustifyRight           | NewHFWOccurrence       |
| KeepTextTogetherDlg    | NumberingOptionDlg     |
| KerningSpacing         | OLEAcceptLinks         |
| KeyboardSelectDlg      | OLECancelLink          |
| LabelsDlg              | OLECancelLinks         |
| Language               | OLEChangeLink          |
| LanguageDlg            | OLEInitializeLinks     |
| LastOpenedAdd          | OLEInsertDlg           |
| LineHeightDlg          | OLEInsertObject        |
| LineHyphenationDlg     | OLELinksDlg            |
| LineNumberingDlg       | OLEObject              |
| LineSpacingDlg         | OLEPasteDlg            |
| LinkUpdate             | OLESetLinkUpdate       |
| ListDefineDlg          | OLEUpdateLink          |
| MacroCommandInserter   | OutlineBarDisplay      |
| MacroEditControlBar    | OutlineBodyTextToggle  |
| MacroEditDlg           | OutlineDefineDlg       |
| MacroExecuteQuick      | OutlineFamilyCut       |
| MacroFilePlay          | OutlineNumberDlg       |
| MacroMenuAdd           | OutlineOff             |
| MacroMenuDelete        | PageBorderDlg          |
| MacroPause             | PageNumberingDlg       |
| MacroStatusPrompt      | PageNumberSettingsDlg  |
| MacroTemplatePlay      | PageSuppressDlg        |
| MailDlg                | PageZoomFull           |
| MarginReleaseInsert    | PaneNext               |
| MarkCrossRef           | PanePrevious           |
| MarkCrossRefUpdateList | PaperSizeAdd           |
| MarkIndex              | PaperSizeDelete        |

|                                |                         |
|--------------------------------|-------------------------|
| PaperSizeDlg                   | PrefPrintSettingsDlg    |
| PaperSizeResave                | PrefPrintSizeRatio      |
| ParagraphBorderDlg             | PrefRevealCodes         |
| ParagraphFormatDlg             | PrefRevealCodesDlg      |
| ParagraphHangingIndent         | PrefRuler               |
| ParagraphNumberDefDlg          | PrefSave                |
| PasteLink                      | PrefStatusBar           |
| PosGoToDlg                     | PrefStatusBarEdit       |
| PosLineVeryBeg                 | PrefStatusBarOptions    |
| PosParagraphNext               | PrefStatusBarOptionsDlg |
| PosParagraphPrevious           | PrefToA                 |
| PosTableBegin                  | PrefUnitsOfMeasure      |
| PosTableColumnBottom           | PrefZoom                |
| PosTableColumnTop              | PrinterCommandDlg       |
| PosTableRowBegin               | PrinterSelectDlg        |
| PowerBarShow                   | QuickFormat             |
| PrefBackupDlg                  | QuickFormatDlg          |
| PrefBackupPath                 | QuickFormatToggle       |
| PrefBeep                       | Redisplay               |
| PrefButtonBar                  | ResetKeyboardMenu       |
| PrefDisplayDlg                 | RulerBarShow            |
| PrefDisplaySet                 | RulerShow               |
| PrefDocSummary                 | SaveAll                 |
| PrefDocSummaryDlg              | ScrollScreenLeft        |
| PrefEnvironment                | ScrollScreenRight       |
| PrefEnvSettingsDlg             | SelectAll               |
| PrefHyphenation                | SelectCell              |
| PrefImportDlg                  | SelectCellDown          |
| PrefImportOptions              | SelectCellDownArrow     |
| PrefLiteCodes                  | SelectCellLeft          |
| PrefLocationDlg                | SelectCellRight         |
| PrefLocationOfDocuments        | SelectCellUp            |
| PrefLocationOfGraphicsFiles    | SelectCellUpArrow       |
| PrefLocationOfHyphenationFiles | SelectCharNext          |
| PrefLocationOfMacroFiles       | SelectCharPrevious      |
| PrefLocationOfPrinterFiles     | SelectColumnBottom      |
| PrefLocationOfSpreadsheetFiles | SelectColumnNext        |
| PrefLocationOfTemplates        | SelectColumnPrevious    |
| PrefMenu                       | SelectColumnTop         |
| PrefMergeCodes                 | SelectDocBottom         |
| PrefMetafileOptions            | SelectDocTop            |
| PrefPowerBar                   | SelectDocVeryTop        |
| PrefPowerBarEdit               | SelectLineBegin         |
| PrefPowerBarFontFaceList       | SelectLineDown          |
| PrefPowerBarFontSizeList       | SelectLineEnd           |
| PrefPrintCopies                | SelectLineUp            |
| PrefPrintDoc                   | SelectLineVeryBegin     |
| PrefPrintRedline               | SelectLineVeryEnd       |

|                              |                         |
|------------------------------|-------------------------|
| SelectMode                   | TableAutoFill           |
| SelectPage                   | TableCalculateDlg       |
| SelectPageNext               | TableConvertDlg         |
| SelectPagePrevious           | TableCopyFormula        |
| SelectParagraph              | TableCreateNewDlg       |
| SelectParagraphNext          | TableDataFill           |
| SelectParagraphPrevious      | TableDeleteDlg          |
| SelectSave                   | TableDeleteTable        |
| SelectScreenDown             | TableDisableCellLocks   |
| SelectScreenLeft             | TableFormulaCBar        |
| SelectScreenRight            | TableGridLinesToggle    |
| SelectScreenUp               | TableGuides             |
| SelectSentence               | TableInsertDlg          |
| SelectSentenceNext           | TableInsertOneRow       |
| SelectSentencePrevious       | TableJoinCells          |
| SelectTable                  | TableLineFillDlg        |
| SelectTableColumn            | TableNameCreateDlg      |
| SelectTableColumnExtendLeft  | TableNumberTypeDlg      |
| SelectTableColumnExtendRight | TablePointMode          |
| SelectTableRow               | TableQuickSum           |
| SelectToggle                 | TableSelectOff          |
| SelectWord                   | TableSelectOn           |
| SelectWordNext               | TableSplitCellsDlg      |
| SelectWordPrevious           | TabSetDlg               |
| SetCancel                    | TemplateCopyObjectDlg   |
| SetDefaultFileFormat         | TemplateCreate          |
| SetDocumentMapping           | TemplateCreateDlg       |
| SettingsDlg                  | TemplateDescriptionDlg  |
| SetupWritingToolsDlg         | TemplateDlg             |
| ShowBoxes                    | TemplateEdit            |
| ShowGraphics                 | TemplateSelect          |
| ShowHiddenText               | TextBoxCreate           |
| ShowSymbols                  | TOACreateFullFormDlg    |
| SoftSpace                    | TOAEditFullFormCBar     |
| SoundClipDlg                 | TOCDefine               |
| SoundClipPlayNext            | TOCDefineDlg            |
| SoundClipSetAttributes       | TypesetManualKerningDlg |
| SoundClipSetDescription      | VarAssign               |
| SoundClipSetFilename         | VarDiscard              |
| StatusBarShow                | ViewDraft               |
| SubdividePageDlg             | ViewOutline             |
| TabInsert                    | ViewPage                |
| TableAdjustBegin             | ViewTwoPage             |
| TableAdjustColumnWidth       | VLineCreate             |
| TableAdjustEnd               | WatermarkDlg            |
| TableAdjustPosition          | WatermarkPlacement      |
| TableAppendRow               | WatermarkPlacementDlg   |
| TableAutoCalc                | WordCountDlg            |

WordLetterspaceDlg  
WPActivate  
WPCharactersDlg  
WPDraw  
WPGrammatik  
WPSpeller  
WPThesaurus  
WritingToolsDisable  
ZoomDlg  
ZoomToFullPage  
ZoomToMarginWidth  
ZoomToPageWidth



# Appendix D: Macro Conversion

## Conversion from 5.2 to 6.0

To convert a WordPerfect 5.2 for Windows macro to 6.0 format,

- 1 Choose **File, Open**, enter a macro filename, then **OK**.
- 2 Choose **File, Preferences**, double-click the **Button Bar** icon, select **Macro Tools**, then choose **Select**.
- 3 Choose **Macro Conv** to display the **5.2 Macro Converter** dialog box.
- 4 Options:
  - ♦ To pause before each conversion and accept or edit a suggested command replacement, select **Prompt for Replacement**.
  - ♦ To convert all macros that require conversion to a default command, select **Replace All**.
  - ♦ To create a backup copy (.BAK) of the original macro, select **Backup Original Macro**.
- 5 Choose **Start**.

Converted 5.2 commands appear as comments (//) in the new macro file.

WordPerfect 6.0 equivalents for 5.2 macros that cannot be converted also appear as comments (//).

The following WordPerfect 5.2 commands can be converted to 6.0 format:

- ♦ Advance
- ♦ BlockProtect
- ♦ BoxEditCaption
- ♦ BoxFigureEdit
- ♦ BoxNewNumber
- ♦ BoxOptions
- ♦ ButtonBarAddItem
- ♦ ButtonBarDeleteItem
- ♦ ButtonBarSelect
- ♦ ButtonBarShow
- ♦ ColumnDefine
- ♦ ColumnsOn
- ♦ CommentConvertToText
- ♦ CommentCreate
- ♦ ConvertClipboardDlg
- ♦ ConvertClipboardPicture
- ♦ DateFormat
- ♦ DDECreateLink
- ♦ DDEEditLink
- ♦ DisplayPitch
- ♦ DocCompare
- ♦ DocSummarySaveAs
- ♦ EditPaste
- ♦ EndnoteOptions
- ♦ ExecuteFunction
- ♦ FileChangeDir
- ♦ Font
- ♦ FootnoteOptions
- ♦ GraphicLine
- ♦ GraphicLineSelect
- ♦ IndexMark
- ♦ KeyboardSelect
- ♦ LineHeight
- ♦ LineNumbering
- ♦ ListDefine
- ♦ ListMark
- ♦ MacroMenuAppend
- ♦ MacroMenuDelete
- ♦ MasterDocCondense
- ♦ OutlineDefine
- ♦ OverstrikeCreate
- ♦ OverstrikeEdit
- ♦ PaperSizeAdd
- ♦ PaperSizeDelete
- ♦ PaperSizeSelect
- ♦ PrefDisplaySet

- ♦ PrefDocSummary
- ♦ PrefEquations
- ♦ PrefHyphenation
- ♦ PrefPrintDoc
- ♦ PrefRevealCodeColors
- ♦ PrefToA
- ♦ PrefZoom
- ♦ PrinterCommand
- ♦ PrinterSelectListFree
- ♦ PrinterSelectListGetItem
- ♦ PrintMultiplePages
- ♦ Sort
- ♦ StylesEdit
- ♦ StylesRetrieve
- ♦ TableCell
- ♦ TableColumn
- ♦ TableConvert
- ♦ TableDeleteColumn
- ♦ TableDeleteRow
- ♦ TableEditColumn
- ♦ TableFormula
- ♦ TableFormulaDlg
- ♦ TableInsertColumn
- ♦ TableInsertRow
- ♦ TableJoin
- ♦ TableLines
- ♦ TableOptions
- ♦ TableRow
- ♦ TableSplit
- ♦ TabSet
- ♦ TextBoxCaptionEdit
- ♦ TextBoxRotate
- ♦ ToCDefine
- ♦ ToCMark
- ♦ TypeChar
- ♦ TypesetDlg
- ♦ Undelete
- ♦ UserBoxCaptionEdit

**Conversion from WP6.0  
for DOS to 6.0 for  
Windows**

Some WordPerfect 6.0 for Windows product commands have the same name, but different parameters or parameter names than their DOS counterparts. The following table displays the variations.

| Command Name              |     | Parameters                                                            | Num.   | Enumerated Type |
|---------------------------|-----|-----------------------------------------------------------------------|--------|-----------------|
| AttributeAppearanceOff    | DOS | Attribute                                                             |        |                 |
|                           | WIN | Attrib                                                                | 16     | Every!          |
| AttributeAppearanceOn     | DOS | Attribute                                                             |        |                 |
|                           | WIN | Attrib                                                                | 16     | Every!          |
| AttributeAppearanceToggle | DOS | Attribute                                                             |        |                 |
|                           | WIN | Attrib                                                                | 16     | Every!          |
| BackupOriginalDoc         | DOS |                                                                       | 0<br>1 | No!<br>Yes!     |
|                           | WIN |                                                                       | 0<br>1 | Off!<br>On!     |
| BorderColor               | DOS | Color Name<br>Red Value<br>Green Value<br>Blue Value<br>Shading Value |        |                 |
|                           | WIN | ColorName<br>Red<br>Green<br>Blue<br>Shade                            |        |                 |
| BorderStyleCopy           | DOS | Style Name<br><br>Current Location<br>Destination<br>New Name         |        |                 |
|                           | WIN | Style<br><br>FromLibrary<br>ToLibrary<br>DestName                     |        |                 |
| BoxCreate                 | DOS | Box Style                                                             |        |                 |
|                           | WIN | BoxType                                                               |        |                 |
| BoxDelete                 | DOS | Box Number                                                            |        |                 |

| Command Name             | Parameters |                                     | Num.                                                                                                  | Enumerated Type                                                                                                                                                                                                                                                                                |
|--------------------------|------------|-------------------------------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BoxDelete (cont.)        | WIN        | BoxType                             | 0<br>1<br>2<br>3<br>4<br>5<br>6<br>7<br>127                                                           | FigureBox!<br>TableBox!<br>TextBox!<br>UserBox!<br>EquationBox!<br>ButtonBox!<br>WatermarkImageBox!<br>InlineEquationBox!<br>NoBox!                                                                                                                                                            |
|                          |            | BoxNumber                           |                                                                                                       |                                                                                                                                                                                                                                                                                                |
| BoxEditByCounter         | DOS        | Counter<br>Number                   |                                                                                                       |                                                                                                                                                                                                                                                                                                |
|                          | WIN        | CounterName<br>Word                 |                                                                                                       |                                                                                                                                                                                                                                                                                                |
| BoxEditNext              | DOS        | Box Type                            |                                                                                                       |                                                                                                                                                                                                                                                                                                |
|                          | WIN        | BoxType                             |                                                                                                       |                                                                                                                                                                                                                                                                                                |
| BoxEditPrevious          | DOS        | Box Type                            |                                                                                                       |                                                                                                                                                                                                                                                                                                |
|                          | WIN        | BoxType                             |                                                                                                       |                                                                                                                                                                                                                                                                                                |
| BoxEquationSaveAsGraphic | DOS        |                                     | 730<br>738                                                                                            | Bitmap!<br>Presentations_20!                                                                                                                                                                                                                                                                   |
|                          | WIN        |                                     |                                                                                                       |                                                                                                                                                                                                                                                                                                |
| BoxImageRetrieve         | DOS        | Image Location<br>Format (optional) | 725<br>726<br>727<br>728<br>729<br>730<br>731<br>732<br>733<br>734<br>736<br>737<br>738<br>739<br>741 | WordPerfectGraphic_10!<br>WordPerfectGraphic_20!<br>TaggedImageFileFormat!<br>PCPaintbrushX!<br>WindowsMetaFile!<br>Bitmap!<br>CMG!<br>AutoCAD_DXF!<br>EncapsulatedPostScript!<br>HewlettPackardGraphicPlotter!<br>Macintosh_PICT!<br>MicroGraftx!<br>Presentations_20!<br>LotusPIC!<br>TarGA! |
|                          | WIN        | Action                              |                                                                                                       |                                                                                                                                                                                                                                                                                                |

| Command Name    |     | Parameters                                            | Num.                                                        | Enumerated Type                                                                                                                                                                                                       |
|-----------------|-----|-------------------------------------------------------|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BoxImageSave    | DOS | Format (optional)                                     | 725<br>726<br>730<br>738                                    | WordPerfectGraphic_10!<br>WordPerfectGraphic_20!<br>Bitmap!<br>Presentations_20!                                                                                                                                      |
|                 | WIN |                                                       |                                                             |                                                                                                                                                                                                                       |
| DeleteAppend    | DOS | Filename (optional)                                   |                                                             |                                                                                                                                                                                                                       |
|                 | WIN |                                                       |                                                             |                                                                                                                                                                                                                       |
| DisplayZoom     | DOS | State                                                 | 0<br>1<br>2                                                 | ZoomMarginWidth!<br>ZoomPageWidth!<br>ZoomFullPage!                                                                                                                                                                   |
|                 | WIN | Percent                                               |                                                             |                                                                                                                                                                                                                       |
| DocSummaryPrint | DOS | Filename (optional)                                   |                                                             |                                                                                                                                                                                                                       |
|                 | WIN |                                                       |                                                             |                                                                                                                                                                                                                       |
| FileCopy        | DOS | Source File<br>Destination File<br>Subdocs (optional) | 1                                                           | Subdocs!                                                                                                                                                                                                              |
|                 | WIN | SourceFile<br>DestinationFile                         |                                                             |                                                                                                                                                                                                                       |
| FileDelete      | DOS | Subdocs (optional)                                    | 1                                                           | Subdocs!                                                                                                                                                                                                              |
|                 | WIN |                                                       |                                                             |                                                                                                                                                                                                                       |
| FileOpen        | DOS | File Format (optional)                                | 3<br>20<br>21<br>74<br>75<br>173<br>800                     | WordPerfect_51/52!<br>WordPerfectMac_20!<br>WordPerfectMac_21!<br>WordForWindows_2A!<br>WordForWindows_2B!<br>Kermit!<br>SpreadsheetDIF!                                                                              |
|                 | WIN | Format (optional)                                     | 3<br>74<br>75<br>82<br>83<br>84<br>161<br>173<br>190<br>191 | WordPerfect_51!<br>WordForWindows_20a!<br>WordForWindows_20b!<br>DisplayWrite_40!<br>DisplayWrite_42!<br>DisplayWrite_50!<br>ANSIText_Windows!<br>Kermit_7bit_Transfer!<br>XyWriteIIIPlus_355!<br>XyWriteIIIPlus_356! |

| Command Name       | Parameters |                                    | Num.                                                         | Enumerated Type                                                                                                                                                                                              |
|--------------------|------------|------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FileOpen (cont.)   |            |                                    | 311<br>312<br>313<br>314<br>315<br>656<br>657<br>807         | AmiPro_12!<br>AmiPro_12A!<br>AmiPro_12B!<br>AmiPro_20!<br>AmiPro_30!<br>ANSIDelimitedTextWindows!<br>ANSIGenericWordProcessor_Windows!<br>Excel_40!                                                          |
| FootnoteEdit       | DOS        | Footnote Number                    |                                                              |                                                                                                                                                                                                              |
|                    | WIN        | FootnoteNumber                     |                                                              |                                                                                                                                                                                                              |
| FootnoteNewNumber  | DOS        | Number                             |                                                              |                                                                                                                                                                                                              |
|                    | WIN        | FootnoteNumber                     |                                                              |                                                                                                                                                                                                              |
| GraphicsLineDelete | DOS        | Graphics Line Number               |                                                              |                                                                                                                                                                                                              |
|                    | WIN        | Line                               |                                                              |                                                                                                                                                                                                              |
| KeyboardSelect     | DOS        | Filename<br>Equation (optional)    | 4                                                            | Equation!                                                                                                                                                                                                    |
|                    | WIN        | Keyboard                           |                                                              |                                                                                                                                                                                                              |
| LineSegmentSetWPG2 | DOS        | Pattern Set<br>Line Pattern Number |                                                              |                                                                                                                                                                                                              |
|                    | WIN        | Style                              | 0<br>1<br>2<br>3<br>4<br>5<br>6<br>7<br>8<br>9<br>126<br>127 | SingleLine!<br>DoubleLine!<br>DashedLine!<br>DottedLine!<br>ThickLine!<br>ExtraThickLine!<br>ThinThickLine!<br>ThickThinLine!<br>ButtonTopLeftLine<br>ButtonBottomRightLine!<br>TableDefaultLine!<br>NoLine! |
|                    |            | LineStyle                          |                                                              |                                                                                                                                                                                                              |
| ListDefinition     | DOS        | Name                               |                                                              |                                                                                                                                                                                                              |
|                    | WIN        | ListName                           |                                                              |                                                                                                                                                                                                              |

| Command Name           | Parameters | Num. Enumerated Type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ListDefinition (cont.) | Style      | 0 NormalStyle!<br>16 FootnoteNumberInDocumentStyle!<br>17 EndnoteNumberInDocumentStyle!<br>18 FigureNumberStyle!<br>19 TableBoxNumberStyle!<br>20 TextBoxNumberStyle!<br>21 UserBoxNumberStyle!<br>22 EquationNumberStyle!<br>23 HypertextStyle!<br>24 ToC1Style!<br>25 ToC2Style!<br>26 ToC3Style!<br>27 ToC4Style!<br>28 ToC5Style!<br>29 Index1Style!<br>30 Index2Style!<br>31 ListStyle!<br>32 ToAStyle!<br>33 DocStyle!<br>34 BoxTextStyle!<br>35 CaptionStyle!<br>36 FootnoteStyle!<br>37 EndnoteStyle!<br>38 CommentStyle!<br>39 HeaderAStyle!<br>40 HeaderBStyle!<br>41 FooterAStyle!<br>42 FooterBStyle!<br>43 WatermarkAStyle!<br>44 WatermarkBStyle!<br>45 ParagraphOutInStyle!<br>46 OutlineStyle!<br>47 LegalOutInStyle!<br>48 BulletsOutInStyle!<br>49 HeadingsOutInStyle!<br>50 Legal_2OutlineStyle!<br>51 NumbersOutInStyle!<br>52 Level1Style!<br>53 Level2Style!<br>54 Level3Style!<br>55 Level4Style!<br>56 Level5Style!<br>57 Level6Style!<br>58 Level7Style!<br>59 Level8Style!<br>60 Legal1Style! |

| Command Name           |     | Parameters                | Num.  | Enumerated Type          |
|------------------------|-----|---------------------------|-------|--------------------------|
| ListDefinition (cont.) | WIN | Numbering                 | 0     | DefNone!                 |
|                        |     |                           | 1     | DefFollows!              |
|                        |     |                           | 2     | DefParentheses!          |
|                        |     |                           | 3     | DefFlushRight!           |
|                        |     |                           | 4     | DefLeader!               |
|                        |     | Counter                   | 0     | FigureBox!               |
|                        |     |                           | 1     | TableBox!                |
|                        |     |                           | 2     | TextBox!                 |
|                        |     |                           | 3     | UserBox!                 |
|                        |     |                           | 4     | EquationBox!             |
|                        |     |                           | 5     | ButtonBox!               |
|                        |     |                           | 6     | WatermarkImageBox!       |
|                        |     |                           | 7     | InlineEquationBox!       |
|                        |     |                           | 127   | NoBox!                   |
|                        |     | NumberingFormat           |       |                          |
| MatchWithAttributes    | DOS | Attribute Name (optional) |       |                          |
|                        | WIN |                           |       |                          |
| MatchWithFont          | DOS | Font Name (optional)      |       |                          |
|                        | WIN | Typeface (optional)       |       |                          |
|                        |     | Family                    | 256   | FamilyITCBenguiat!       |
|                        |     |                           | 2304  | FamilyTimesRoman!        |
|                        |     |                           | 4352  | FamilyITCGaramond!       |
|                        |     |                           | 6400  | FamilyCenturySchoolbook! |
|                        |     |                           | 8448  | FamilyBodon!             |
|                        |     |                           | 10496 | FamilyITCLubalinGraph!   |
|                        |     |                           | 12544 | FamilyAlbertus!          |
|                        |     |                           | 14592 | FamilyCourier!           |
|                        |     |                           | 16640 | FamilyHelvetica!         |
|                        |     |                           | 18688 | FamilyITCRonda!          |
|                        |     |                           | 20736 | FamilyMicrostyle!        |
|                        |     |                           | 22784 | FamilyOptima!            |
|                        |     |                           | 24832 | FamilyLetterGothic!      |
|                        |     |                           | 26880 | FamilyBrush!             |
|                        |     |                           | 28928 | FamilyParkAvenue!        |
|                        |     |                           | 30976 | FamilySignetRoundhand!   |
|                        |     |                           | 33024 | FamilyTekton!            |
|                        |     |                           | 35072 | FamilyUncial!            |
|                        |     |                           | 37120 | FamilyOldEnglish!        |
|                        |     |                           | 39168 | FamilyITCZapfDingbars!   |



| Command Name          | Parameters | Num.                 | Enumerated Type            |
|-----------------------|------------|----------------------|----------------------------|
| MatchWithFont (cont.) | Attributes | 41216                | FamilyBroadway!            |
|                       |            | 43264                | FamilyComputer!            |
|                       |            | 45312                | FamilyPTBarnum!            |
|                       |            | 47360                | FamilyPonderosa!           |
|                       |            | 49408                | FamilyMadrone!             |
|                       |            | 51456                | FamilyHobo!                |
|                       |            | 53504                | FamilyRevue!               |
|                       |            | 55552                | FamilyAestheticOrnamented! |
|                       |            | 0                    | FontMatchNormal!           |
|                       |            | 1                    | FontMatchItalic!           |
|                       | 2          | FontMatchOutline!    |                            |
|                       | 4          | FontMatchShadow!     |                            |
|                       | 8          | FontMatchSmallCaps!  |                            |
|                       | Weight     | 0                    | WeightUnknown!             |
|                       |            | 1                    | WeightUltraThin!           |
|                       |            | 8                    | WeightUltraLight!          |
|                       |            | 16                   | WeightExtraThin!           |
|                       |            | 24                   | WeightThin!                |
|                       |            | 32                   | WeightExtraLight!          |
|                       |            | 48                   | WeightLight!               |
|                       |            | 64                   | WeightDemiLight!           |
|                       |            | 80                   | WeightSemiLight!           |
|                       |            | 88                   | WeightBook!                |
|                       |            | 96                   | WeightRegular!             |
|                       |            | 104                  | WeightRoman!               |
|                       |            | 112                  | WeightMedium!              |
|                       |            | 128                  | WeightDemiBold!            |
|                       |            | 136                  | WeightSemiBold!            |
|                       |            | 144                  | WeightBold!                |
|                       |            | 160                  | WeightExtraBold!           |
|                       |            | 176                  | WeightHeavy!               |
|                       |            | 192                  | WeightExtraHeavy!          |
|                       |            | 208                  | WeightBlack!               |
|                       | 224        | WeightExtraBlack!    |                            |
|                       | 240        | WeightUltraHeavy!    |                            |
|                       | 248        | WeightUltraBlack!    |                            |
|                       | 254        | WeightMaximum!       |                            |
|                       | Width      | 0                    | WidthUnknown!              |
|                       |            | 16                   | WidthUltraCondensed!       |
|                       |            | 32                   | WidthUltraCompressed!      |
|                       |            | 48                   | WidthCompressed!           |
| 64                    |            | WidthExtraCondensed! |                            |

| Command Name          | Parameters                                                                                                                                                                     | Num.         | Enumerated Type     |                              |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|---------------------|------------------------------|
| MatchWithFont (cont.) |                                                                                                                                                                                | 80           | WidthCondensed!     |                              |
|                       |                                                                                                                                                                                | 96           | WidthSemiCondensed! |                              |
|                       |                                                                                                                                                                                | 112          | WidthNormal!        |                              |
|                       |                                                                                                                                                                                | 128          | WidthSemiExpanded!  |                              |
|                       |                                                                                                                                                                                | 144          | WidthExpanded!      |                              |
|                       |                                                                                                                                                                                | 160          | WidthExtraExpanded! |                              |
|                       |                                                                                                                                                                                | 176          | WidthUltraExpanded! |                              |
|                       |                                                                                                                                                                                | 192          | WidthDoubleWide!    |                              |
|                       |                                                                                                                                                                                | 224          | WidthTripleWide!    |                              |
|                       |                                                                                                                                                                                | Source       | 16                  | PRsFile!                     |
|                       |                                                                                                                                                                                |              | 20                  | DRsFile!                     |
|                       |                                                                                                                                                                                | Type         | 134                 | Speedo!                      |
|                       |                                                                                                                                                                                |              | 135                 | Intellifont!                 |
|                       |                                                                                                                                                                                |              | 138                 | Type!                        |
|                       |                                                                                                                                                                                |              | 139                 | TrueType!                    |
|                       |                                                                                                                                                                                | CharacterSet | 0                   | FontMatchASCII!              |
|                       |                                                                                                                                                                                |              | 1                   | FontMatchMultinational1!     |
|                       |                                                                                                                                                                                |              | 2                   | FontMatchMultinational2!     |
|                       |                                                                                                                                                                                |              | 3                   | FontMatchBoxDrawing!         |
|                       |                                                                                                                                                                                |              | 4                   | FontMatchTypographicSymbols! |
|                       |                                                                                                                                                                                |              | 5                   | FontMatchIconicSymbols!      |
|                       |                                                                                                                                                                                |              | 6                   | FontMatchMath!               |
|                       |                                                                                                                                                                                |              | 7                   | FontMatchMathExtension!      |
|                       |                                                                                                                                                                                |              | 8                   | FontMatchGreek!              |
|                       |                                                                                                                                                                                |              | 9                   | FontMatchHebrew!             |
|                       |                                                                                                                                                                                |              | 10                  | FontMatchCyrillic!           |
|                       |                                                                                                                                                                                |              | 11                  | FontMatchJapaneseKana!       |
| 12                    | FontMatchUserDefined!                                                                                                                                                          |              |                     |                              |
| 13                    | FontMatchArabic!                                                                                                                                                               |              |                     |                              |
| 14                    | FontMatchArabicScript!                                                                                                                                                         |              |                     |                              |
| PrintDocOnDisk        | DOS<br>Page Range<br>Secondary Page Range<br>(optional)<br>Chapter Range (optional)<br>Volume Range (optional)<br>Odd/Even Pages (optional)<br>DocSummary<br>Pamphlet<br>Order |              |                     |                              |

| Command Name           |     | Parameters                                                                                                                                                                                                       | Num.                                      | Enumerated Type                                                                                              |
|------------------------|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| PrintDocOnDisk (cont.) | WIN | PageRange<br>SecondaryRange<br>ChapterRange<br>VolumeRange<br>OddEvenPages<br>PrintSummary<br>PrintBooklet<br>Direction                                                                                          |                                           |                                                                                                              |
| PrinterSelect          | DOS | Printer<br>Filename (optional)                                                                                                                                                                                   |                                           |                                                                                                              |
|                        | WIN | PrinterType<br><br>PrinterName<br>PrinterPort                                                                                                                                                                    | 0<br>1                                    | WPPrinter!<br>WindowsPrint!                                                                                  |
| PrintMultiplePages     | DOS | Page Range (optional)<br>Secondary Page Range (optional)<br>Chapter Range (optional)<br>Volume Range (optional)<br>Odd/Even Pages (optional)<br>DocSummary (optional)<br>Pamphlet (optional)<br>Order (optional) |                                           |                                                                                                              |
|                        | WIN | PageRange<br>SecondaryPageRange<br>ChapterRange<br>VolumeRange<br>OddEvenPages<br>PrintSummary<br>PrintBooklet<br>Direction                                                                                      |                                           |                                                                                                              |
| ReplaceWithAttributes  | DOS | Attribute Name                                                                                                                                                                                                   |                                           |                                                                                                              |
|                        | WIN | Attribute                                                                                                                                                                                                        | 0<br>1<br>2<br>3<br>4<br>5<br>6<br>7<br>8 | ExtraLarge!<br>VeryLarge!<br>Large!<br>Small!<br>Fine!<br>Superscript!<br>Subscript!<br>Outline!<br>Italics! |

| Command Name                     | Parameters |                | Num.                                                                                                                                                                                                                                               | Enumerated Type                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------|------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReplaceWithAttributes<br>(cont.) |            |                | 9                                                                                                                                                                                                                                                  | Shadow!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                                  |            |                | 10                                                                                                                                                                                                                                                 | Redline!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|                                  |            |                | 11                                                                                                                                                                                                                                                 | DoubleUnderline!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|                                  |            |                | 12                                                                                                                                                                                                                                                 | Bold!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|                                  |            |                | 13                                                                                                                                                                                                                                                 | Strikeout!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                  |            |                | 14                                                                                                                                                                                                                                                 | Underline!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                  |            |                | 15                                                                                                                                                                                                                                                 | SmallCaps!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                                  |            | 16             | Every!                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ReplaceWithFont                  | DOS        | Font Name      |                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|                                  | WIN        | Name<br>Family | 256<br>2304<br>4352<br>6400<br>8448<br>10496<br>12544<br>14592<br>16640<br>18688<br>20736<br>22784<br>24832<br>26880<br>28928<br>30976<br>33024<br>35072<br>37120<br>39168<br>41216<br>43264<br>45312<br>47360<br>49408<br>51456<br>53504<br>55552 | FamilyITCBenguiat!<br>FamilyTimesRoman!<br>FamilyITCGaramond!<br>FamilyCenturySchoolbook!<br>FamilyBodon!<br>FamilyITCLubalinGraph!<br>FamilyAlbertus!<br>FamilyCourier!<br>FamilyHelvetica!<br>FamilyITCRonda!<br>FamilyMicrostyle!<br>FamilyOptima!<br>FamilyLetterGothic!<br>FamilyBrush!<br>FamilyParkAvenue!<br>FamilySignetRoundhand!<br>FamilyTekton!<br>FamilyUncial!<br>FamilyOldEnglish!<br>FamilyITCZapfDingbats!<br>FamilyBroadway!<br>FamilyComputer!<br>FamilyPTBarnum!<br>FamilyPonderosa!<br>FamilyMadrone!<br>FamilyHobo!<br>FamilyRevue!<br>FamilyAestheticOrnamented! |
|                                  | Attributes |                | 0<br>1<br>2<br>4<br>8                                                                                                                                                                                                                              | FontMatchNormal!<br>FontMatchItalic!<br>FontMatchOutline!<br>FontMatchShadow!<br>FontMatchSmallCaps!                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

| Command Name            | Parameters          | Num.  | Enumerated Type   |     |                       |
|-------------------------|---------------------|-------|-------------------|-----|-----------------------|
| ReplaceWithFont (cont.) | Weight              | 0     | WeightUnknown!    |     |                       |
|                         |                     | 1     | WeightUltraThin!  |     |                       |
|                         |                     | 8     | WeightUltraLight! |     |                       |
|                         |                     | 16    | WeightExtraThin!  |     |                       |
|                         |                     | 24    | WeightThin!       |     |                       |
|                         |                     | 32    | WeightExtraLight! |     |                       |
|                         |                     | 48    | WeightLight!      |     |                       |
|                         |                     | 64    | WeightDemiLight!  |     |                       |
|                         |                     | 80    | WeightSemiLight!  |     |                       |
|                         |                     | 88    | WeightBook!       |     |                       |
|                         |                     | 96    | WeightRegular!    |     |                       |
|                         |                     | 104   | WeightRoman!      |     |                       |
|                         |                     | 112   | WeightMedium!     |     |                       |
|                         |                     | 128   | WeightDemiBold!   |     |                       |
|                         |                     | 136   | WeightSemiBold!   |     |                       |
|                         |                     | 144   | WeightBold!       |     |                       |
|                         |                     | 160   | WeightExtraBold!  |     |                       |
|                         |                     | 176   | WeightHeavy!      |     |                       |
|                         |                     | 192   | WeightExtraHeavy! |     |                       |
|                         |                     | 208   | WeightBlack!      |     |                       |
|                         |                     | 224   | WeightExtraBlack! |     |                       |
|                         |                     | 240   | WeightUltraHeavy! |     |                       |
|                         |                     | 248   | WeightUltraBlack! |     |                       |
|                         |                     | 254   | WeightMaximum!    |     |                       |
|                         |                     | Width | Width             | 0   | WidthUnknown!         |
|                         |                     |       |                   | 16  | WidthUltraCondensed!  |
|                         |                     |       |                   | 32  | WidthUltraCompressed! |
|                         |                     |       |                   | 48  | WidthCompressed!      |
|                         |                     |       |                   | 64  | WidthExtraCondensed!  |
|                         |                     |       |                   | 80  | WidthCondensed!       |
|                         |                     |       |                   | 96  | WidthSemiCondensed!   |
|                         |                     |       |                   | 112 | WidthNormal!          |
|                         |                     |       |                   | 128 | WidthSemiExpanded!    |
| 144                     | WidthExpanded!      |       |                   |     |                       |
| 160                     | WidthExtraExpanded! |       |                   |     |                       |
| 176                     | WidthUltraExpanded! |       |                   |     |                       |
| 192                     | WidthDoubleWide!    |       |                   |     |                       |
| 224                     | WidthTripleWide!    |       |                   |     |                       |
| Source                  | Source              | 16    | PRSFFile!         |     |                       |
|                         |                     | 20    | DRSFFile!         |     |                       |

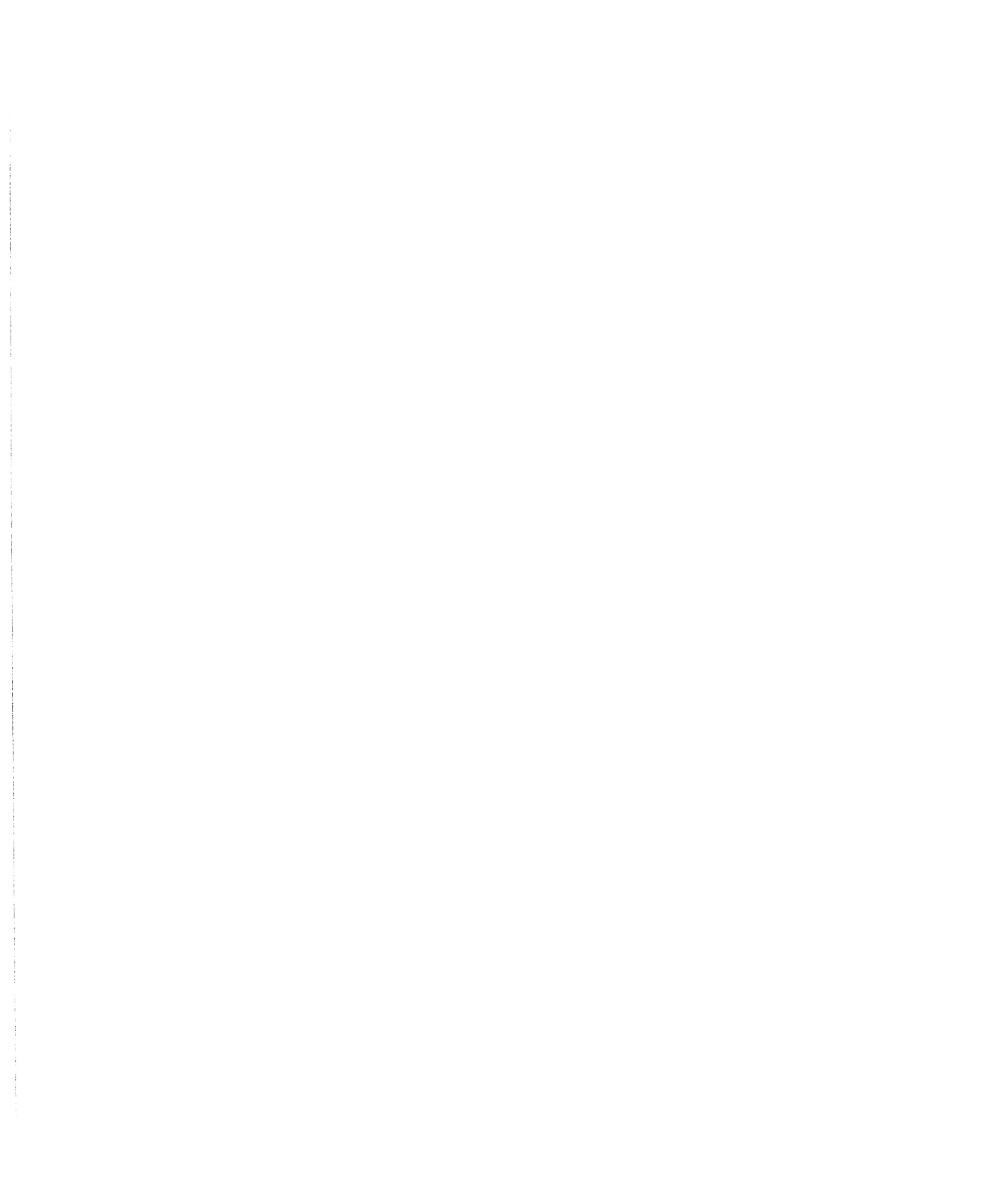
| Command Name            | Parameters |                                                                | Num. | Enumerated Type              |
|-------------------------|------------|----------------------------------------------------------------|------|------------------------------|
| ReplaceWithFont (cont.) |            | Type                                                           | 134  | Speedo!                      |
|                         |            |                                                                | 135  | Intellifont!                 |
|                         |            |                                                                | 138  | Type1!                       |
|                         |            |                                                                | 139  | TrueType!                    |
|                         |            | CharacterSet                                                   | 0    | FontMatchASCII!              |
|                         |            |                                                                | 1    | FontMatchMultinational1!     |
|                         |            |                                                                | 2    | FontMatchMultinational2!     |
|                         |            |                                                                | 3    | FontMatchBoxDrawing!         |
|                         |            |                                                                | 4    | FontMatchTypographicSymbols! |
|                         |            |                                                                | 5    | FontMatchIconicSymbols!      |
|                         |            |                                                                | 6    | FontMatchMath!               |
|                         |            |                                                                | 7    | FontMatchMathExtension!      |
|                         |            |                                                                | 8    | FontMatchGreek!              |
|                         |            |                                                                | 9    | FontMatchHebrew!             |
|                         |            |                                                                | 10   | FontMatchCyrillic!           |
|                         |            |                                                                | 11   | FontMatchJapaneseKana!       |
|                         |            |                                                                | 12   | FontMatchUserDefined!        |
|                         |            |                                                                | 13   | FontMatchArabic!             |
|                         |            |                                                                | 14   | FontMatchArabicScript!       |
| ReplaceWithFontSize     | DOS        | Font Size                                                      |      |                              |
|                         | WIN        | FontSize                                                       |      |                              |
| Sort                    | DOS        | Source (optional)<br>Destination (optional)<br>Undo (optional) |      |                              |
|                         | WIN        | InFile (optional)<br>OutFile (optional)<br>State (optional)    |      |                              |
| StyleCodes              | DOS        | Off Codes State                                                |      |                              |
|                         | WIN        | State<br>Library                                               | 0    | CurrentDoc!                  |
|                         |            |                                                                | 1    | PersonalLibrary!             |
|                         |            |                                                                | 2    | SharedLibrary!               |
| StyleSystemOn           | DOS        | Name                                                           |      |                              |
|                         | WIN        | Style<br>Action                                                | 0    | NormalStyle!                 |
|                         |            |                                                                | 1    | SkipEmptyParagraphs!         |
| TableAutoFormulaEntry   | DOS        |                                                                | 0    | Off!                         |
|                         |            |                                                                | 1    | On!                          |

| Command Name                     | Parameters |                                                                                                                              | Num.                                                                      | Enumerated Type                                                                                                                                                                                                                                                                |
|----------------------------------|------------|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TableAutoFormulaEntry<br>(cont.) | WIN        |                                                                                                                              | 0<br>1                                                                    | No!<br>Yes!                                                                                                                                                                                                                                                                    |
| TableCellLineColor               | DOS        | Color Name (optional)<br>Red Value (optional)<br>Green Value (optional)<br>Blue Value (optional)<br>Shading Value (optional) |                                                                           |                                                                                                                                                                                                                                                                                |
|                                  | WIN        | ColorName<br>Red<br>Green<br>Blue<br>Shade                                                                                   |                                                                           |                                                                                                                                                                                                                                                                                |
| TableDefaultLineColor            | DOS        | Color Name<br>Red Value<br>Green Value<br>Blue Value<br>Shading Value                                                        |                                                                           |                                                                                                                                                                                                                                                                                |
|                                  | WIN        | ColorName<br>Red<br>Green<br>Blue<br>Shade                                                                                   |                                                                           |                                                                                                                                                                                                                                                                                |
| ToADefinition                    | DOS        | Section Name                                                                                                                 |                                                                           |                                                                                                                                                                                                                                                                                |
|                                  | WIN        | SectionName                                                                                                                  |                                                                           |                                                                                                                                                                                                                                                                                |
|                                  |            | Style                                                                                                                        | 0<br>16<br>17<br>18<br>19<br>20<br>21<br>22<br>23<br>24<br>25<br>26<br>27 | NormalStyle!<br>FootnoteNumberInDocumentStyle!<br>EndnoteNumberInDocumentStyle!<br>FigureNumberStyle!<br>TableBoxNumberStyle!<br>TextBoxNumberStyle!<br>UserBoxNumberStyle!<br>EquationNumberStyle!<br>HypertextStyle!<br>ToC1Style!<br>ToC2Style!<br>ToC3Style!<br>ToC4Style! |

| Command Name          | Parameters | Num. | Enumerated Type      |
|-----------------------|------------|------|----------------------|
| ToADefinition (cont.) |            | 28   | ToC5Style!           |
|                       |            | 29   | Index1Style!         |
|                       |            | 30   | Index2Style!         |
|                       |            | 31   | ListStyle!           |
|                       |            | 32   | ToAStyle!            |
|                       |            | 33   | DocStyle!            |
|                       |            | 34   | BoxTextStyle!        |
|                       |            | 35   | CaptionStyle!        |
|                       |            | 36   | FootnoteStyle!       |
|                       |            | 37   | EndnoteStyle!        |
|                       |            | 38   | CommentStyle!        |
|                       |            | 39   | HeaderAStyle!        |
|                       |            | 40   | HeaderBStyle!        |
|                       |            | 41   | FooterAStyle!        |
|                       |            | 42   | FooterBStyle!        |
|                       |            | 43   | WatermarkAStyle!     |
|                       |            | 44   | WatermarkBStyle!     |
|                       |            | 45   | ParagraphOutlnStyle! |
|                       |            | 46   | OutlineStyle!        |
|                       |            | 47   | LegalOutlnStyle!     |
|                       |            | 48   | BulletsOutlnStyle!   |
|                       |            | 49   | HeadingsOutlnStyle!  |
|                       |            | 50   | Legal_2OutlineStyle! |
|                       |            | 51   | NumbersOutlnStyle!   |
|                       |            | 52   | Level1Style!         |
|                       |            | 53   | Level2Style!         |
|                       |            | 54   | Level3Style!         |
|                       |            | 55   | Level4Style!         |
|                       |            | 56   | Level5Style!         |
|                       |            | 57   | Level6Style!         |
|                       |            | 58   | Level7Style!         |
|                       |            | 59   | Level8Style!         |
|                       |            | 60   | Legal1Style!         |
|                       |            | 61   | Legal2Style!         |
|                       |            | 62   | Legal3Style!         |
|                       |            | 63   | Legal4Style!         |
|                       |            | 64   | Legal5Style!         |
|                       |            | 65   | Legal6Style!         |
|                       |            | 66   | Legal7Style!         |
|                       |            | 67   | Legal8Style!         |
|                       |            | 68   | Heading1Style!       |
|                       |            | 69   | Heading2Style!       |
|                       |            | 70   | Heading3Style!       |
|                       |            | 71   | Heading4Style!       |
|                       |            | 72   | Heading5Style!       |
|                       |            | 73   | Heading6Style!       |



| Command Name                                                          |     | Parameters                   | Num.       | Enumerated Type |
|-----------------------------------------------------------------------|-----|------------------------------|------------|-----------------|
| ToADefinition (cont.)                                                 |     | Numbering                    | 0          | DefNone!        |
|                                                                       |     |                              | 1          | DefFollows!     |
|                                                                       |     |                              | 2          | DefParentheses! |
|                                                                       |     |                              | 3          | DefFlushRight!  |
|                                                                       |     | 4                            | DefLeader! |                 |
|                                                                       |     | Underline                    | 0          | NoToAUnderline! |
|                                                                       |     |                              | 1          | ToAUnderline!   |
|                                                                       |     | PageCombine                  | 0          | NoCombine!      |
|                                                                       |     |                              | 1          | Combine!        |
|                                                                       |     | NumberingFormat              |            |                 |
| ToAEditFullForm                                                       | DOS | Short Form                   |            |                 |
|                                                                       | WIN | ShortForm                    |            |                 |
| ToAEditPageFormat                                                     | DOS | Section Name<br>Page Format  |            |                 |
|                                                                       | WIN | Listname<br>NumberFormat     |            |                 |
| UserFunction (Windows<br>product command, DOS<br>programming command) | DOS | Action                       |            |                 |
|                                                                       | WIN | Action<br>Data               |            |                 |
| WatermarkA                                                            | DOS | State<br>Pages (optional)    |            |                 |
|                                                                       | WIN | Action<br>Occurrence         | 0          | NoPages!        |
| WatermarkB                                                            | DOS | State<br>Pages (optional)    |            |                 |
|                                                                       | WIN | Action<br>Occurrence         | 0          | NoPages!        |
| WordLetterSpacing                                                     | DOS | Word<br>Letter               |            |                 |
|                                                                       | WIN | CharacterWidth<br>SpaceWidth |            |                 |



# Glossary/Index





# Glossary

- active** An application, window, or dialog box that is currently in use. Windows highlights the caption or dialog frame of the active window.
- ANSI character set** The 256 characters of the American National Standards Institute. Used by WordPerfect for Windows.
- application** A software program, such as a word processing or spreadsheet program.
- argument** A variable, constant, or expression required by a command or function.
- ASCII character set** The American Standard Code for Information Interchange is one of the standard formats for representing characters. The ASCII character set contains 128 characters, which are the first 128 characters of the ANSI character set.
- BIF (Binary Initialization File)** A file that contains run-time options for a WordPerfect application. BIF replaces the standard text initialization files (INI) commonly used by Windows applications. See *run-time* below.
- bitmap** A graphics file format that represents images as a pattern of *pixels* (dots).
- button** A graphic representation of a command or option.
- character expression** One or more characters enclosed in quotation marks, which identifies the characters as text, not a variable.
- check box** A graphic representation where the user turns an option on or off by selecting or deselecting the box. An X in the box indicates the option is on.
- choose** To click a menu item or dialog box option that initiates an immediate action. See also *select*.
- click** To press and release the mouse button.
- Clipboard** An area of memory, also called a buffer, where text and commands can be stored for further action. Clipboard contents are erased when you exit Windows.
- codes** Formatting commands that specify computer and printer functions. Reveal Codes displays codes.
- color wheel** A dialog box graphic where the user selects a color.
- command** An instruction given to a computer.
- command button** A button in a dialog box which carries out an action OK, Exit, or Cancel. A thick, dark border indicates the default button.
- compile** To translate program code into machine language and check for syntax errors. The program automatically compiles macros the first time they are played. Each time a macro is edited and saved, it must be recompiled before it can be played.
- conditional statement** A statement created with macro programming commands such as IF and WHILE. A conditional statement contains expressions such as text, numbers, or variables to be evaluated. Macro execution can be affected by the "condition" of these expressions at the time they are evaluated.

**control menu** The menu used to open, close, maximize, minimize, or restore a window or dialog box. To display a control menu, click the control-menu box or press **Alt+Spacebar**.

**control-menu box** A small rectangular button in the upper left corner of a window or dialog box. Click a control-menu box to display the control menu; double-click to close the window or dialog box. Several windows can be open at the same time, each with its own control-menu box.

**counter** A dialog box element where the user specifies values with incrementer and decrementer buttons.

**CUA keyboard** Common User Access (CUA) standard keyboard. The WordPerfect for Windows default keyboard is CUA compliant.

**current cell, current file, current table** The cell, document, or table where the insertion point is.

**current directory** The directory where a file will be saved if you do not specify another.

**DDE (Dynamic Data Exchange)** A connection between a WordPerfect document and a file created by another Windows application that supports DDE. You can transfer information from one file to the other through the link. If both files are open, transferred information can be updated automatically when information changes in the source file. See also *spreadsheet link*.

**default** A setting, value, or response provided by the application unless an alternative is specified. You can change some WordPerfect defaults for all future documents in Preferences. Other options change a default for the active document only.

**desktop** The screen background and environment for all Windows applications.

**dialog box** A window that displays warnings and messages, and options for the user to select. Dialog boxes have a title bar and a control menu, but not a menu bar. They can be moved to different positions on the screen. Most dialog boxes must be closed before you can work in the document window, but a few (such as Edit Button Bar) allow you to move between the window and the dialog box.

**dialog units** A vertical unit that equals one-eighth of the font height, or a horizontal unit that equals one-fourth of the font width. Dialog boxes are positioned and sized in dialog units.

**directory** A group of files or directories under one identifying title.

The *root directory* is the directory on a disk drive to which all other directories are related.

A *default directory* is where files are saved to or opened and retrieved from if you do not specify a pathname. Not all features in WordPerfect have the same default directory; many defaults can be customized.

A *directory tree* is a diagram of how directories and subdirectories relate to each other.

**DOS** Disk Operating System. Software that directs the flow of data between disk drives and your computer. An application needs an operating system to function.

**double-click** To click the mouse button twice in rapid succession.

**drag** Press and hold down the left mouse button while moving the mouse.

**driver** A set of commands that runs a peripheral device such as a printer or monitor.

**edit box** An element in a dialog box where a user can type text.

**enumerated type** The parameters for some product commands accept only certain predefined words known as “enumerated types.” Enumerated types end with exclamation points and have numeric equivalents. For example, DisplayMode parameter accepts only Text!, Graphics!, or FullPage!.

**extension** The optional three-letter identifier that may be added to any filename. The default document extension used by WordPerfect 6.0 for Windows is .WPD.

**file** A document or other type of information saved under an identifying name.

**file format** The pattern in which a file is organized. Each application creates documents in a unique format generally not accessible by other applications. Those source documents must be converted to the new application’s format.

**filename** The name given to a file on disk. The name may be up to eight characters long, with an optional three letter extension, such as MYLETTER.ANN.

**group** A set of related options in a dialog box, often with its own subtitle.

**.INI file** A file that stores setup and startup information for Windows applications.

**incrementing button** A button that lets you specify an amount by clicking the mouse instead of typing numbers.

**input focus** The window or control receiving Windows messages has input focus. Common controls are radio buttons, push buttons, and edit boxes.

**insertion point** The blinking vertical bar in the current document window or dialog box. It indicates where text will appear when you type.

**kerning** Increasing or decreasing space between a pair of characters.

**link** see *DDE* and *spreadsheet link*.

**list box** A dialog box control that displays a list of mutually-exclusive items.

**loop** A set of programming commands that repeat.

**macro** A series of commands and keystrokes written and/or recorded in its own file to be compiled, then replayed as needed.

**macro command** An instruction for WordPerfect to carry out. The instruction can be in the form of a programming command or a product command. Many macro commands must be used together with other macro commands to be successful in giving a complete instruction.

**measurement expression** A number followed by a unit of measure (c, i p w, “).

**memory** The temporary data storage area for a computer or printer.

**menu** A list of commands that can be applied to the active window or application.

**message box** A dialog box that displays information, a warning, or error message, or asks for confirmation before completing a command.

**numeric equivalent** The number that represents an enumerated type.

**numeric expression** A number on which mathematical operations may be performed. Numeric expressions are not enclosed in quotation marks.

**OLE (Object Linking and Embedding)** A feature that copies information from one document to another (embedding), through a “live” link. When the original document changes, the embedded copy reflects the changes.

**operator** A symbol or word that performs a function on one or more expressions. Operators compare expressions, link words together, and perform mathematical functions.

**parameter** Contains data to pass to the compiler or pass between subroutines.

**path** The location of a certain file or directory in a computer disk drive or on a network. A path includes the drive, the root directory, and any subdirectory names. Each name is separated by a backslash (\). For example, C:\WPWIN60\LETTER refers to the LETTER subdirectory in the WPWIN60 directory on the C drive.

**pixel** The smallest unit of display on a computer screen.

**point size** The unit of measure for font sizes. One point equals 1/72”.

**popup list** A list of options displayed when a popup button is clicked. Most popup buttons are marked by double triangles and display mutually exclusive options. The button itself shows the selected option. Other popup lists, marked by single triangles, show the feature name rather than the selected option. On the Ruler Bar, popup list buttons have no triangular markings. Some display the selected option and others display the feature name.

**port** A connection device between a computer and another component such as a printer or modem.

**printer driver** Software commands that enable an application to communicate with a printer. WordPerfect can use WordPerfect printer drivers and Windows printer drivers. WordPerfect printer drivers are identified by a .PRS extension.

**product prefix** A two-character expression that specifies a product for a macro command.

**prompt** A message box that displays information for the user.

**radio button** Represents one item in a list of mutually exclusive options.

**relational expression** An expression that evaluates parameters with only two possible states: True and False.



**Reveal Codes** A feature that displays text and formatting codes in the document window.

**root directory** See *directory*.

**run-time** Execution time. Run-time errors occur during macro execution. Run-time options are start up settings such as the macros default directory. See *BIF* above.

**scroll** To move through a document or list box by using the scroll bar.

**scroll bar** The vertical or horizontal bar in a document and in some list boxes that allows you to move through a document or list by clicking the scroll arrows or dragging the scroll box (thumb).

**select** To identify a file, directory, graphics box, or area of text that will be affected by subsequent choices; to identify a dialog box option to be applied to a file, directory, graphics box, or area of text.

**selection cursor** The highlighted text, dotted rectangle, or insertion point that indicates where the next keystroke or mouse action will apply in a dialog box.

**Shared Code** Programming code used by WordPerfect applications for Windows.

**sizing handle** The small solid squares that appear on the borders of a selected graphics box. Drag handles to size the box and its contents.

**sort key** Words, fields, or phrases that define sort criteria.

**spreadsheet import** Copies information from a spreadsheet file into a WordPerfect document one time only. You cannot update a spreadsheet import like you can update a spreadsheet link.

**spreadsheet link** A connection to transfer information between a spreadsheet file and a WordPerfect document. You can update transferred information to reflect changes made in the spreadsheet file. See also *DDE link*.

**status bar** The line at the bottom of the WordPerfect window that lists the current font and the position of the insertion point. The status bar also displays some WordPerfect messages and features.

**syntax** The order in which parts of a command are organized. Macro commands require a specific syntax to function properly.

**text box (dialog box)** The area in a dialog box where you type text or where WordPerfect types text for you. The insertion point must be in the text box before you can begin to type.

**text box (graphics)** A type of graphics box for placing and rotating text.

**text file** A file saved in ASCII (DOS) or ANSI (Windows) file format. It contains text, spaces, and returns, but no formatting codes.

**title bar** The area of a window or dialog box that displays its title.

**toggle** To switch between on or off; or, a feature or option that is turned on or off with the same keystroke or command. Toggle commands usually do not require parameters.

**user-defined dialog box** A custom dialog box created with DIALOG programming commands that display options for user input.

**value set member** Words specified for acceptance by certain parameters. Each value set member ends with an exclamation point (!). Every value set member has a numeric equivalent.

**variable** One or more characters that represent data that can change during macro execution.

**wildcard** Characters that represent variables in a word, file, or directory search. A question mark (?) represents a single character. An asterisk (\*) represents zero or more characters in succession.

**window** Each window has a title bar, a menu bar, and a status bar, and may contain a scroll bar, Button Bar, and Ruler Bar. The Equation Editor window has separate editing and display areas called *panes*.

**window handle** A unique identifier for a window or control.

// (Comment) 73

## A

Abbreviation

- in template, creating 241
- in template, deleting 241
- replacing with full word(s) 242

AbbreviationCreate 241

AbbreviationDelete 241

AbbreviationDlg 242

AbbreviationExpand 242

AboutDlg 243

Accelerator text, assign 449

Activating a window 73

Advance 243

AdvanceDlg 244

?Align 867

Aligning currency symbols in tables  
781, 809

Aligning text *see also* Columns; Tables

- at left margin 499, 500
- at right margin 499, 500
- between left/right margins 499, 500
- centering between margins 499
- centering one line 337
- hanging indent 615
- justification options, specify 499, 931
- on both margins 499

Alphabetize *see* Sort

APPACTIVATE 73

Appearance attributes *see also* Text  
appearance

- turning off 246
- turning on 247

Appearance of boxes *see* BoxStyle  
commands

Append 244

APPEXECUTE 74

APPEXECUTEEXT 74

APPLICATION 76

- default, specifying a new 193
- ending 160
- locating 77
- starting 74
- using in a macro 76

Application statement 37

APPLOCATE 77

AppMaximize 244

AppMinimize 244

AppMove 245

AppRestore 245

AppSize 246

AppTaskList 246

Arithmetic Expressions 21

Arrange alphabetically/numerically *see*  
Sort

Arrays

- create/assign to persist variable table  
206
- creating persist 207
- global variable, creating 174
- local variable, creating 104, 183

ASCII

- beginning/end of field, specify 484
- beginning/end of record, specify 484
- encapsulation character, specify 483
- extra characters, removing in import  
485

ASCII file, speaking 190

ASSERT 44, 77

ASSIGN 79

Assign Macro to Menu dialog box,  
displaying 536

Assigning a macro 69

Assignment operator, := 79

Assignment Statements 7

creating 79

AssociateDlg 246

Attribute

- font size, specifying 250
- of blocked text, turning on/off 248
- position, toggling 250
- positioning text vertically 249
- relative font, determining size of 251
- relative size, toggling 251
- relative size, turning on/off 251
- restoring font to normal 249
- turning off 246
- turning on 247

AttributeAppearanceOff 246

AttributeAppearanceOn 247

AttributeAppearanceToggle 248

AttributeNormal 249

AttributePosition 249

AttributePositionToggle 250

AttributeRelativeSize 250

AttributeRelativeSizeToggle 251

AttributeSizeRatios 251

Audio *see* Sound clip

Authorities *see* Table of Authorities

Automatic Kerning, turning on/off 501

?AutoUpdate 867

## B

Background for graphics boxes/tables *see*  
the Fill commands

BackTab 252

Backup 252

- location of files, specifying 646
- minutes between, specifying 253
- Original Document Backup, turning  
on/off 253

path for files, specifying 637

BackupMinutes 253

BackupOriginalDoc 253

BarcodeDlg 254

BarcodePOSTNET 254

Baseline Placement for Typesetting,  
turning on/off 255

?BaselineMode 867

BEEP 80

Beep, specify default setting 638

?BetweenTableCodes 867

BIF *see* Binary Initialization File

BIFFILEPATH 81

BIFINFO 81

BIFREAD 83

BIFWRITE 85

Binary digits 5

Binary Initialization File

- information from, retrieving 81
- path and filename, retrieving 81
- reading 83
- writing to 85

Binary Operators 13

Binding 255

?Binding 868

BindingOptionsDlg 256

Bitwise Expressions 23

Bitwise Operators 16

Block *see* the Select commands

?BlockActive 868

BlockProtect 256

?BlockProtect 868

BoldKey 257

Bookmark

- QuickMark, finding 675
- QuickMark, setting 675

BookmarkBlock 257

BookmarkCreate 257

- BookmarkDelete 258
- BookmarkDlg 258
- BookmarkFind 259
- ?BookmarkList 869
- BookmarkMove 259
- BookmarkRename 260
- Border color
  - blue, specifying amount of 870
  - for current border, specifying 277
  - for paragraph/page/column, specifying 261
  - green, specifying amount of 870
  - option, using 874
  - red, specifying amount of 870
  - shading percentage, specifying 871
- Border spacing
  - Automatic Spacing option, selecting 270
  - bottom inside, specifying 871
  - inside, specifying 265
  - outside, specifying 267
  - setting 873
  - top inside, specifying 872
- Border style *see also* Style, border
  - copying 270
  - corner radius, specifying 871
  - deleting 272
  - editing 273
  - editing, end 274
  - fill color, specifying 264
  - fill style, specifying 829
  - left line, specifying 872
  - naming new 272
  - renaming 275
  - retrieving 275
  - right line, specifying 873
  - saving 276
  - shadow, specifying location/size/color 263
  - specifying 830
  - top line, specifying 874
- Border, line style
  - bottom line, specifying 869
  - bottom, specifying 260
  - corners, customizing curve 262
  - left, specifying 266
  - right, specifying 268
  - separator in paragraph/columns, specifying 269
  - separator line, specifying 873
  - top, specifying 276
- Border, text *see* Text Border
- ?Border 869
- BorderBottomLine 260
- ?BorderBottomLine 869
- BorderColor 261
- ?BorderColorBlue 870
- ?BorderColorGreen 870
- ?BorderColorRed 870
- ?BorderColorShading 871
- BorderCornerRadius 262
- ?BorderCornerRadius 871
- BorderDropShadow 263
- BorderFillColors 264
- BorderInsideSpacing 265
- ?BorderInsideSpacingBottom 871
- ?BorderInsideSpacingTop 872
- BorderLeftLine 266
- ?BorderLeftLine 872
- BorderOutsideSpacing 267
- BorderRightLine 268
- ?BorderRightLine 873
- BorderSeparatorLine 269
- ?BorderSeparatorLine 873
- BorderSetSpacing 270
- ?BorderSetSpacing 873
- BorderStyleCopy 270
- BorderStyleCreate 272
- BorderStyleDelete 272
- BorderStyleEdit 273
- BorderStyleEnd 274
- BorderStyleName 275
- BorderStyleRetrieve 275
- BorderStyleSave 276
- BorderTopLine 276
- ?BorderTopLine 874
- BorderUseBorderColor 277
- ?BorderUseBorderColor 874
- Box
  - black and white threshold, specifying 305
  - border style, specifying 279
  - current number, displaying 892
  - current version, displaying 329
  - editing next occurrence of 294
  - editing previous occurrence of 295
  - editing specific 293
  - fill style, specifying 303
  - fill values, resetting 318
  - number of in current document 880
  - position values, resetting 318
  - reset values for border, specifying 316
  - saving/cancelling changes 295
  - text line height, specifying 878
  - type and number, specifying 293
- alignment, specifying 876
- closing 280
- default number style, specifying 875
- editing 279
- formatting 876
- initial style default, specifying 875
- initial style for, specifying 281
- location of, customizing 284
- numbering style, specifying 283
- position offset, specifying 877
- position relative to border, specifying 877
- reset values, specifying 316
- rotating 286, 878
- side position, specifying 877
- width, specifying 280
- Box content
  - creating 288
  - editing 288
  - editing using OLE server 314
  - filename, specifying 878
  - horizontal position of, specifying 879
  - positioning 288
  - preserving aspect ratio 879
  - reset values, specifying 316
  - resetting content render values 317
  - storing in document/on disk, specify 879
  - type, specifying 290, 880
  - vertical position of, specifying 880
- Box counter
  - reset values, specifying 317
  - type, assigning new 290
  - type, specifying 881
- Box equation
  - blue, specifying amount of 881
  - color/shade values of, specifying 296
  - default font, selecting 297
  - filename, specifying 298
  - font size, specifying 301, 883
  - font, specifying 298, 883
  - green, specifying amount of 881
  - red, specifying amount of 882
  - redisplaying 301
  - saving in graphics file format 302
  - shading 882
- Box image
  - background colors/gradients, showing 313
  - black and white threshold, specifying 885
  - brightness, specifying 306, 886
  - colors, inverting 309, 889

- contrast, specifying 307, 886
- converting to black and white 310
- dither method, specifying 886
- dither source, specifying 887
- fill type, specifying 307, 887
- halftone angle, specifying 888
- halftone screen LPI, specifying 889
- horizontal position of, specifying 891
- monochrome, specifying 889
- page background, specifying 890
- retrieving 311
- reversing on X axis 308, 887
- reversing on Y axis 309, 888
- rotating 312, 890
- saving 312
- scaling 313
- scaling height, specifying 891
- scaling width, specifying 890
- vertical position of, specifying 891
- within graphics box, moving 310
- Box style *see also* Style
  - copying 319
  - creating 320
  - deleting 320
  - editing 321
  - modifying when editing/creating 287
  - naming 322, 893
  - retrieving 323
  - saving all 323
  - saving/cancelling changes 322
- Box text
  - closing text box 327
  - contouring around image 329, 896
  - contouring to square shape 329
  - default style, specifying 325, 894
  - flow direction, specifying 896
  - flow values, resetting 318
  - flow/placement of text, specifying 328
  - line height, specifying 286
  - naming text file 327
  - opening text box 327
  - rotating 324
  - rotation, specifying degree of 894
- Box, position on page
  - aligning with margin/column(s) 884
  - aligning with margins 895
  - alignment, specifying 304
  - attaching to page/move with text 893
  - attaching to page/paragraph/character 278, 875
  - column, aligning with left 892
  - column, aligning with right 893
  - height, specifying 303, 883
  - horizontal alignment, specifying 305, 884
  - horizontal offset, specifying 885
  - overlapping with other boxes 315, 892
  - vertical alignment, specifying 330, 894
  - vertical position relative to alignment, specifying 895
  - vertical position, specifying 330
  - width, setting 331, 895
  - width/height ratio, preserving 289
- BoxAttachTo 278
- ?BoxAttachTo 875
- BoxBorder 279
- BoxBorderDlg 279
- ?BoxCaptionDefaultInitialStyle 875
- ?BoxCaptionDefaultNumberStyle 875
- BoxCaptionEdit 279
- BoxCaptionEditEnd 280
- BoxCaptionFormatting 280
- ?BoxCaptionFormatting 876
- BoxCaptionInitialStyle 281
- BoxCaptionNumberStyle 283
- BoxCaptionPosition 284
- ?BoxCaptionPositionAlignment 876
- ?BoxCaptionPositionBorder 877
- ?BoxCaptionPositionOffset 877
- ?BoxCaptionPositionSide 877
- BoxCaptionRotation 286
- ?BoxCaptionRotation 878
- BoxChangeLineHeight 286
- ?BoxChangeLineHeight 878
- BoxChangeStyle 287
- BoxContentEdit 288
- ?BoxContentFilename 878
- ?BoxContentHorizontalPosition 879
- ?BoxContentInternal 879
- BoxContentPosition 288
- BoxContentPreserveAspectRatio 289
- ?BoxContentPreserveAspectRatio 879
- BoxContentType 290
- ?BoxContentType 880
- ?BoxContentVerticalPosition 880
- ?BoxCount 880
- BoxCounter 290
- ?BoxCounter 881
- BoxCreate 291
- BoxDelete 292
- BoxEdit 293
- BoxEditByCounter 293
- BoxEditNext 294
- BoxEditPrevious 295
- BoxEnd 295
- BoxEquationColor 296
- ?BoxEquationColorBlue 881
- ?BoxEquationColorGreen 881
- ?BoxEquationColorRed 882
- ?BoxEquationColorShading 882
- BoxEquationDefaultFont 297
- BoxEquationFilename 298
- BoxEquationFont 298
- ?BoxEquationFont 883
- BoxEquationFontSize 301
- ?BoxEquationFontSize 883
- BoxEquationRedisplay 301
- BoxEquationSaveAsGraphic 302
- Boxes around text/graphics *see the* Border commands
- BoxFill 303
- BoxHeight 303
- ?BoxHeight 883
- BoxHorizontalAlignment 304
- ?BoxHorizontalAlignment 884
- ?BoxHorizontalAlignTo 884
- ?BoxHorizontalOffset 885
- BoxHorizontalPosition 305
- BoxImageBlackWhiteThreshold 305
- ?BoxImageBlackWhiteThreshold 885
- BoxImageBrightness 306
- ?BoxImageBrightness 886
- BoxImageContrast 307
- ?BoxImageContrast 886
- ?BoxImageDitherMethod 886
- ?BoxImageDitherSource 887
- BoxImageFill 307
- ?BoxImageFill 887
- BoxImageFlipX 308
- ?BoxImageFlipX 887
- BoxImageFlipY 309
- ?BoxImageFlipY 888
- ?BoxImageHalftoneAngle0 888
- ?BoxImageHalftoneAngle1 888
- ?BoxImageHalftoneAngle2 888
- ?BoxImageHalftoneAngle3 889
- ?BoxImageHalftoneScreenLPI 889
- BoxImageInvertColors 309
- ?BoxImageInvertColors 889
- BoxImageMonochrome 310
- ?BoxImageMonochrome 889
- BoxImageMove 310
- ?BoxImagePageBackground 890
- BoxImageRetrieve 311
- BoxImageRotate 312
- ?BoxImageRotation 890
- BoxImageSave 312

BoxImageScaling 313  
 ?BoxImageScalingX 890  
 ?BoxImageScalingY 891  
 BoxImageShowBackground 313  
 ?BoxImageTranslationX 891  
 ?BoxImageTranslationY 891  
 BoxInitialStyleDlg 314  
 ?BoxLeftColumn 892  
 ?BoxNumber 892  
 BoxOLEEditObject 314  
 BoxOverlap 315  
 ?BoxOverlap 892  
 ?BoxPageOffset 893  
 BoxPosDlg 315  
 BoxResetBorder 316  
 BoxResetCaption 316  
 BoxResetContent 316  
 BoxResetContentRender 317  
 BoxResetCounter 317  
 BoxResetFill 318  
 BoxResetPosition 318  
 BoxResetTextFlow 318  
 ?BoxRightColumn 893  
 BoxStyleCopy 319  
 BoxStyleCreate 320  
 BoxStyleDelete 320  
 BoxStyleEdit 321  
 BoxStyleEnd 322  
 BoxStyleName 322  
 ?BoxStyleName 893  
 BoxStyleRetrieve 323  
 BoxStyleSave 323  
 BoxTextAngle 324  
 ?BoxTextAngle 894  
 BoxTextDefaultStyle 325  
 ?BoxTextDefaultStyle 894  
 BoxTextEdit 327  
 BoxTextEditEnd 327  
 BoxTextFilename 327  
 BoxTextFlow 328  
 BoxTextFlowContour 329  
 BoxTextFlowSquare 329  
 BoxUpdateDisplay 329  
 BoxVerticalAlignment 330  
 ?BoxVerticalAlignment 894  
 ?BoxVerticalAlignTo 895  
 ?BoxVerticalOffset 895  
 BoxVerticalPosition 330  
 BoxWidth 331  
 ?BoxWidth 895  
 BoxWrapDlg 332  
 ?BoxWrapping 896  
 ?BoxWrappingType 896

BREAK 88  
 Buffers available for Undelete 961  
 Bullet, insert 497  
 BulletDlg 332  
 Button bar  
   displaying 336  
   font, specifying 333  
   hiding 474  
   hiding/showing 336  
   location, specifying 334  
   rows to show, specifying 334  
   selecting specific 333  
   style, specifying 335  
 ?ButtonBarActive 897  
 ButtonBarCreateDlg 332  
 ButtonBarEditDlg 332  
 ButtonBarOptionsDlg 333  
 ButtonBarSelect 333  
 ButtonBarSetFont 333  
 ButtonBarSetLocation 334  
 ButtonBarSetRows 334  
 ButtonBarSetStyle 335  
 ButtonBarShow 336  
 ButtonBarShowToggle 336

## C

Calculations *see* Formulas  
 CALL 28, 89  
   CASE CALL 93  
   ONCANCEL CALL 199  
   ONDDEADWISE CALL 200  
   ONERROR CALL 202  
   ONNOTFOUND CALL 204  
 Callback function, creating 54, 148  
 Calling statements 28  
   CALL 89  
   CASE CALL 93  
   GO 175  
 CANCEL 90  
 Cancel condition  
   asserting 77  
   calling a Label statement 199  
   executing a Label statement 198  
   responding to 90  
 Capitalization, changing *see* Convert case  
 Captions in boxes *see* Box caption  
 CASE 25, 92  
 CASE CALL 93  
 Case, converting *see* Convert case  
 CASEOF, SWITCH-ENDSWITCH 228

CaseToggle 336  
 Cell *see also* Floating cell; Table cell  
   fill, specifying 898  
   formula, specifying 899  
   ignore when calculating 899  
   justification, specifying 899  
   locking 900  
 Cell attribute  
   aligning 897  
   Bold, activating 897  
   cell-specific, specifying 902  
   Double Underline 898  
   Extra Large 898  
   Fine 898  
   Italics 899  
   Large 900  
   Normal 900  
   Outline 900  
   Redline 901  
   Shadow 901  
   Small 901  
   Small Caps 901  
   Strikeout 902  
   Subscript 902  
   Superscript 903  
   Underline 903  
   Very Large 903  
 Cell, floating *see* FloatingCell  
   commands  
   ?Cell 897  
   ?CellAlignment 897  
   ?CellBold 897  
   ?CellDoubleUnderline 898  
   ?CellExtraLarge 898  
   ?CellFill 898  
   ?CellFine 898  
   ?CellFormula 899  
   ?CellIgnoreWhenCalculating 899  
   ?CellItalics 899  
   ?CellJustification 899  
   ?CellLarge 900  
   ?CellLocked 900  
   ?CellNormal 900  
   ?CellOutline 900  
   ?CellRedline 901  
   ?CellShadow 901  
   ?CellSmall 901  
   ?CellSmallCaps 901  
   ?CellSpecificAttribute 902  
   ?CellSpecificJustification 902  
   ?CellStrikeout 902  
   ?CellSubscript 902  
   ?CellSuperscript 903

- ?CellUnderline 903
- ?CellVeryLarge 903
- Center 337
  - all pages vertically 338
  - current page 337
  - one line of text 337
- CenterCurrentPage 337
- ?CenterCurrentPage 903
- CenterPageDlg 337
- CenterPages 338
- ?CenterPages 904
- CHAIN 94
- Chapter number
  - decreasing by one 339
  - entering at insertion point 339
  - increasing by one 339
  - numbering method, specifying 340
  - specifying 338
- ChapterNumber 338
- ?ChapterNumber 904
- ChapterNumberDecrement 339
- ChapterNumberDisplay 339
- ChapterNumberIncrement 339
- ChapterNumberMethod 340
- ?ChapterNumberMethod 904
- Char 340
- Character Expressions 20
- Character functions
  - CHARLEN 96
  - CHARPOS 97
  - SUBCHAR 226
- Characters
  - converting to decimal equivalents 98
  - converting to lower 230
  - converting to upper 231
- Characters, combining *see the*
  - Overstrike commands
- CHARLEN 96
- CharMappingDlg 341
- CHARPOS 97
- ?CharSpaceWidth 904
- Chart, creating with Draw 452
- Check box control, adding to a dialog
  - box 107
- ClearDoc 341
- Clipboard
  - appending current document to 341
  - contents of, copying to current
    - document 391
  - contents, retrieving 342
  - copying and replacing 391
  - deleting and appending 371
  - deleting and copying to 391
  - speaking a file saved to 191
  - text/graphics, saving 342
- ClipboardAppendTo 341
- ClipboardRetrieve 342
- ClipboardSaveTo 342
- Close
  - current window/text editor 342
  - document without saving 343
  - Graphics Control Bar 343
- CloseGraphicsControlBar 343
- CloseNoSave 343
- Coach
  - commands 970
  - dialog box 970
  - overview 969
- CoachAnimate 972
- CoachFilterAdd 973
- CoachFilterDestroy 977
- CoachFilterDisable 978
- CoachFilterEnable 977
- CoachGetRegionInfo 978
- CoachMessageBox 979
- CoachMessageBoxClose 983
- CoachPrompt 983
- CoachPromptClose 985
- CoachRemoveDialogFilter 972
- CoachSetDialogFilter 970
- Code page, specifying preferences 344
- CodePage 344
- CodePageDlg 345
- Codes
  - showing *see* Reveal Codes
  - specifying preferences 645
- Color
  - border fill, specifying 264
  - border, specifying 261
  - border, specifying current 277
  - graphics *see* Graphics line color
  - intensity of, specifying 832
  - modifying/renaming 347
  - text, specifying 831
- Color palette
  - adding color to 345
  - creating 347
  - deleting 348
  - deleting color from 346
  - renaming 349
  - selecting 349
- Color wheel control, adding to a dialog
  - box 108
- ColorAdd 345
- ColorDelete 346
- ColorModify 347
- ColorPaletteCreate 347
- ColorPaletteDelete 348
- ColorPaletteDlg 348
- ColorPaletteRename 349
- ColorPaletteSelect 349
- ?Column 905
- Column(s), table *see* Table column
- Column(s), text
  - blocked text in, converting to table
    - 789
  - break, inserting 801
  - defining 350
  - hard column break, inserting 465
  - inserting 803
  - page break, inserting 465
  - turning off 351
- Column(s), text, moving within
  - bottom of column, moving to 621
  - first column, moving to 622
  - last column, moving to 622
  - next column, moving to 622
  - previous column, moving to 623
  - top of current, moving to 623
- ?ColumnBorder 905
- ColumnBorderDlg 350
- ?ColumnDefine 905
- ColumnDefineDlg 350
- ColumnsDefinition 350
- ColumnsOff 351
- Combination box control, adding to a
  - dialog box 111
- Command Based Macro Language 5
- Command Inserter 36
  - opening/closing in a macro 536
- Command Name 8
- Commands
  - Coach 970
  - CoachAnimate 972
  - CoachFilterAdd 973
  - CoachFilterDestroy 977
  - CoachFilterDisable 978
  - CoachFilterEnable 977
  - CoachGetRegionInfo 978
  - CoachMessageBox 979
  - CoachMessageBoxClose 983
  - CoachPrompt 983
  - CoachPromptClose 985
  - CoachRemoveDialogFilter 972
  - CoachSetDialogFilter 970
  - Product 9
  - Programming 9
- Comment Statements 8, 73
- CommentConvert 351

- CommentCreate 352
  - CommentEdit 352
  - CommentInformation 352
  - CommentInsertTime 352
  - CommentInsertUserInitials 353
  - CommentInsertUserName 353
  - Comments in macros 73
  - Comments in text
    - creating 352
    - editing screen 352
    - text, converting to 351
    - time, inserting current 352
    - user initials, inserting 353
    - user name, inserting 353
  - Compiler 5
  - Compiler errors 6
  - Conditional statements 8, 25
    - CASE 92
    - CASE CALL 93
    - IF-ELSE-ENDIF 176
    - IFPLATFORM-ENDIFPLATFORM 178
    - SWITCH-ENDSWITCH 228
  - ConditionalEndOfPage 353
  - ?ConfirmCodesDelete 906
  - ?ConfirmFormulaDelete 906
  - Constants 11
  - CONTINUE,
    - SWITCH-ENDSWITCH 228
  - Control bar, displaying Hypertext 475
  - Control statements, macro 25
  - Conversions, Radix Base 20
  - Convert *see* the Import commands
  - Convert case
    - initial caps, converting text to 354
    - to lowercase 354
    - to uppercase 354
  - ConvertCaseInitialCaps 354
  - ConvertCaseLowercase 354
  - ConvertCaseUppercase 354
  - Copy 355
  - Counter
    - decreasing 356
    - graphics box reference number, specifying 355
    - increasing 362
    - level type, renaming 360
    - levels, specifying number of 358
    - numbering method for level, specifying 359
    - numbering method, changing 360
    - numbering method, specifying 362
    - user-created, renaming 361
    - value, deleting 356
    - value, displaying 357
    - value, setting 363
  - Counter control, adding to a dialog box 115
  - CounterCreate 355
  - CounterDecrement 356
  - CounterDelete 356
  - CounterDisplay 357
  - CounterDlg 358
  - CounterEditLevel 358
  - CounterEditLevelMethod 359
  - CounterEditLevelName 360
  - CounterEditMethod 360
  - CounterEditName 361
  - CounterIncrement 362
  - CounterSetMethod 362
  - CounterSetValue 363
  - Cross references
    - generating 448
    - marking 542
    - reference mark, specifying 364
    - target name, creating 542
    - target, naming 365
  - CrossRefMarkReference 364
  - CrossRefMarkTarget 365
  - CTON 98
  - Currency symbols
    - align in a block of cells 767
    - align in columns 781
    - align in tables 809
    - type, specifying 785
    - with numbers, displaying 773
  - ?CurrentLabel 906
  - ?CurrentPaperName 906
  - ?CurrentParagraphLevel 906
  - ?CurrentPrinter 907
  - ?CurrentPrinterPort 907
  - ?CurrentPrinterType 907
  - ?CurrentSubdoc 907
  - ?CurrentTableName 908
  - ?CurrentTemplate 908
  - ?CurrentWord 908
  - Cursor movement *see* Insertion point movement; Pos commands
  - CustomBoxDlg 365
- D**
- Date
    - current, inserting 365
    - format, specifying 366
    - system, inserting 367
  - DateCode 365
  - ?DateDay 908
  - DateFormat 366
  - DateFormatDlg 366
  - ?DateMonth 908
  - DateText 367
  - ?DateWeekday 909
  - ?DateYear 909
  - DDE *see* Dynamic Data Exchange
  - DDE link
    - changing 368
    - establishing 367, 369
    - pasting from Clipboard 616
    - removing 368
    - updating 367, 369
  - DDEChangeUpdateMode 367
  - DDECreateLink 367
  - DDEDeleteLink 368
  - DDEEditLink 368
  - DDEEXECUTE 99
  - DDEEXECUTEEXT 100
  - DDEINITIATE 101
  - DDEPasteLink 369
  - DDEPOKE 102
  - DDEREQUEST 102
  - DDETERMINATE 103
  - DDETERMINATEALL 104
  - DDEUpdateLink 369
  - DecimalAlignmentCharacter 370
  - ?DecimalAlignmentCharacter 909
  - DECLARE 11, 104
  - DEFAULT, SWITCH-ENDSWITCH 228
  - Default directory for file types *see* Files, specify location preferences for various
  - Default directory for various file types *see* Path
  - Default document extension, on/off 963
  - ?DefaultDocumentExtension 909
  - ?DefaultTemplateExtension 909
  - DEFAULTUNITS 106
  - ?DefaultView 909
  - DelayCodes 370
  - DelayCodesDlg 371
  - DeleteAppend 371
  - DeleteCharNext 371
  - DeleteCharPrevious 371
  - DeleteToBeginningOfWord 372
  - DeleteToEndOfLine 372
  - DeleteToEndOfPage 372
  - DeleteToEndOfWord 373
  - DeleteWord 373



- Deleting text
  - current word/space 373
  - selected text 699
  - to beginning of word 372
  - to end of line 372
  - to end of page 372
  - to end of word 373
  - to left of insertion point 371
  - to right of insertion point 371
- Dialog box
  - Coach 970
  - creating 145
  - display in macro without pausing 498
  - displaying 148
  - hiding 153
  - modal 42
  - modeless 42
  - removing from memory 147
- Dialog box controls
  - adding a custom control 113
  - DIALOGADDCHECKBOX 107
  - DIALOGADDCOLORWHEEL 108
  - DIALOGADDCOMBOBOX 111
  - DIALOGADDCONTROL 113
  - DIALOGADD COUNTER 115
  - DIALOGADDEDITBOX 118
  - DIALOGADDFILENAMEBOX 120
  - DIALOGADDFRAME 121
  - DIALOGADDGROUPBOX 123
  - DIALOGADDHLINE 125
  - DIALOGADDHOTSPOT 126
  - DIALOGADDICON 127
  - DIALOGADDLISTBOX 129
  - DIALOGADDLISTITEM 131
  - DIALOGADDPOPUPBUTTON 132
  - DIALOGADDPUSHBUTTON 134
  - DIALOGADDRADIOBUTTON 136
  - DIALOGADDSCROLLBAR 137
  - DIALOGADDTEXT 140
  - DIALOGADDVIEWER 142
  - DIALOGADDVLINE 143
- Dialog boxes
  - Abbreviation 242
  - About WordPerfect 243
  - Add Markings 375
  - Associate 246
  - Binding Options 256
  - Bookmark 258
  - Box Border/Fill 279
  - Box Caption 453
  - Box Content 453
  - Box Find 454
  - Box Position 315
  - Box Styles 452
  - Bullets & Numbers 332
  - Button Bar Editor-WordPerfect 332
  - Button Bar Options 333
  - Button Bar Preferences 639
  - Calculate 760
  - Center Page(s) 337
  - Change Outline Level 615
  - Coach 470
  - Code Page Preferences 345
  - Column Border 350
  - Columns 350
  - Comment Information 352
  - Condense/Save Subdocuments 546
  - Convert Table 788
  - Copy/Remove Template Objects 824
  - Counter Numbering 358
  - Create Button Bar 332
  - Create Data File 566
  - Create Data Link 481
  - Create Document Template 824
  - Create Full Form 833
  - Create Graphics Line 509
  - Create Hypertext Link 476
  - Create Merge File 563, 566
  - Create Name 808
  - Create Table 791
  - Custom Box 365
  - Define Color Printing Palette 348
  - Define Index 493
  - Define List 527
  - Define Table of Authorities 833
  - Define Table of Contents 848
  - Delay Codes 371
  - Delete 795
  - Display Preferences 639
  - Display Preferences, reveal codes 659
  - Distance 468
  - Document Character Map 341
  - Document Date/Time Format 366
  - Document Information 860
  - Document Initial Font 379
  - Document Summary 384
  - Document Summary Preferences 641
  - Edit Data Link 483
  - Edit Endnote 393
  - Edit Footnote 441
  - Edit Full Form 843
  - Edit Graphics Line 509
  - Edit Graphics Menu 453
  - Edit Hypertext Link 477
  - Endnote Number 398
  - Endnote Options 397
  - Endnote Placement 398
  - Envelope 399
  - Environment Preferences 643
  - Equation Font 399
  - Expand Master Document 401
  - File Preferences, documents/backup 637
  - File Preferences, location of files 646
  - Find and Replace Text 680
  - Find Text 419
  - Font 433
  - Footnote Number 443
  - Footnote Options 445
  - Force Page 448
  - Format 798
  - Generate 448
  - Go To 625
  - Graphics Styles 464
  - Headers/Footers 468
  - Image Settings 454
  - Import Data 481
  - Import Preferences 644
  - Include Subdocument 547
  - Insert Columns/Rows 803
  - Insert File 405
  - Insert Image 402, 452
  - Insert Merge Codes 559
  - Insert Object 570
  - Keep Text Together 501
  - Keyboard Preferences 503
  - Labels 505
  - Language 508
  - Line Height 510
  - Line Hyphenation 510
  - Line Numbering 513
  - Line Spacing 521
  - Link Options 483
  - Manual Kerning 851
  - Margins 448
  - Menu Bar Preferences 555
  - Merge 560
  - Move 807
  - Number Type 814
  - Numbering Value 608
  - OLELinks 570
  - Open File 407
  - Other Codes 498
  - Outline Define 575
  - Overstrike 602
  - Page Border 603
  - Page Numbering 605

- Page Numbering Options 568
- Paper Size 612
- Paragraph Border 614
- Paragraph Format 615
- Paste Special 571
- Perform Merge 562
- Placement 469, 858
- POSTNET Barcode 254
- Preferences 713
- Print 667
- Print Preferences 657
- Printer Command 669
- QuickFinder 674
- QuickFormat 674
- QuickFormat, toggle 675
- Redline 381
- Remove Markings 376
- Repeat 677
- Retrieve Equation Text 400
- Save As 410
- Search 472
- Select Printer 670
- Set Paragraph Number 581
- Sort (Source and Destination) 717
- Sound Clips 721
- Split Cell 820
- Status Bar Options 663
- Style List 744
- Styles Editor 497
- Subdivide Page 747
- Suppress 608
- Tab Set 823
- Table Lines/Fill 805
- Table Names in Current Document 809
- Template Description 825
- Templates 825
- Text Box Initial Style 314
- Thesaurus 863
- Undelete 851
- Update Import 493
- Watermark 858
- Word Spacing and Letterspacing 860
- WordPerfect Characters 862
- WP Print Process 669
- Wrap Text 332
- Writing Tools 714
- Zoom 864
- Zoom Equation Display 401
- Dialog handle, returning in a variable 152
- DIALOGADDCHECKBOX 107
- DIALOGADDCOLORWHEEL 108
- DIALOGADDCOMBOBOX 111
- DIALOGADDCONTROL 113
- DIALOGADDCOUNTER 115
- DIALOGADDEDITBOX 118
- DIALOGADDFILENAMEBOX 120
- DIALOGADDFRAME 121
- DIALOGADDGROUPBOX 123
- DIALOGADDHLINE 125
- DIALOGADDHOTSPOT 126
- DIALOGADDICON 127
- DIALOGADDLISTBOX 129
- DIALOGADDLISTITEM 131
- DIALOGADDPOPUPBUTTON 132
- DIALOGADDPUSHBUTTON 134
- DIALOGADDRADIOBUTTON 136
- DIALOGADDSCROLLBAR 137
- DIALOGADDTEXT 140
- DIALOGADDVIEWER 142
- DIALOGADDVLINE 143
- DIALOGDEFINE 145
- DIALOGDESTROY 147
- DIALOGDISPLAY 148
- DIALOGHANDLE 152
- DIALOGUNDISPLAY 153
- Directories, default for various file types
  - see* Files, specify location preferences for various
- Directory
  - backup path, specifying 637
  - creating 373
  - documents or graphics, changing 402
  - removing 408
- DirectoryCreate 373
- Directory path *see* Path
- DISCARD 154
- Display 374
- Display preferences
  - formatting codes 645
  - merge codes 652
  - Reveal Codes 658
  - Ruler Bar 660
  - set default 639
  - units of measure 664
- Displaying 374
  - graphics 715
  - graphics boxes 714
  - hidden text 715
  - symbols 715
- ?DisplayMode 910
- ?DisplayPitch 910
- DisplayZoom 374
- DIV 14
- DLL *see* Dynamic Link Library
- DLLCALL 155
- DLLFREE 157
- DLLLOAD 158
- ?DocBlank 910
- ?DocChanged 910
- DocCompare 375
- DocCompareAddMarkingsDlg 375
- DocCompareRemoveMarkingsDlg 376
- DocCompareRemoveRedline 376
- DocInitialFont 376
- DocInitialFontDlg 379
- DocMaximize 380
- DocMinimize 380
- DocMove 380
- DocNext 381
- ?DocNumber 911
- DocPrevious 381
- ?DocReadOnly 911
- DocRedlineMethodDlg 381
- DocRestore 381
- DocSize 382
- DocSummaryConfig 382
- DocSummaryDelete 384
- DocSummaryDlg 384
- DocSummaryPrint 384
- DocSummarySaveAs 385
- DocSumSetFull 386
- Document
  - backup path, specifying 637
  - codes, removing from 376
  - comparing 375
  - default extension on/off 963
  - default font, setting 376
  - generating 448
  - location of files, specifying 646
  - maximizing size to fill screen 380
  - new, opening 405
  - next, activating 381
  - previous size, restoring to 381
  - previous, activating 381
  - repositioning 380
  - shrinking current to icon 380
- Document summary
  - current information, deleting 384
  - current, printing 384
  - default descriptive type, specifying 641
  - fields to include, specifying 382
  - saving 385
  - subject search text, specifying 641
  - user-defined, specifying fields for 386
- Document view *see* View document
- Document window, change size 382

- ?DocumentFont 911
- ?DocumentInformation 911
- dotLeader 388
- ?DotLeader 911
- DoubleSidedPrint 389
- Draft, viewing document without
  - features 853
- DraftZoom100 389
- DraftZoomMargin 390
- Draw
  - opening 862
  - using to create chart 452
- Dynamic Data Exchange 65 *see also* the DDE commands
  - assigning client application data to a variable 102
  - assigning data to a client application item 102
  - ending a DDE conversation 103, 104
  - sending a command string 99, 100
  - starting a DDE conversation 101
  - testing for changes in a DDE link 200
- Dynamic Link Library
  - calling 155
  - loading into memory 158
  - removing from memory 157

## E

- Edit a macro 33
- Edit box control, adding to a dialog box 118
- EditComment 391
- EditCopy 391
- EditCut 391
- EditPaste 391
- ?EditScreen 912
- ELSE 176
- ENDAPP 160
- EndCenterOrAlignment 392
- ENDFOR 163, 164, 166
- ENDIF 176
- ENDIFPLATFORM 178
- Endnote *see also* Footnote
  - creating 392
  - current numbering, displaying 395
  - decreasing 394
  - editing 393
  - generating 448
  - increasing 395
  - new numbering code, inserting 394
  - numbering method, selecting 395
  - options, specifying 396

- placement of, specifying 397
- spacing between 398
- text to keep together, specifying
  - amount of 393
- ?Endnote 912
- EndnoteCreate 392
- EndnoteEditDlg 393
- EndnoteMinimum 393
- ?EndnoteMinimum 912
- EndnoteNewNumber 394
- EndnoteNumberDecrement 394
- EndnoteNumberDisplay 395
- EndnoteNumberIncrement 395
- EndnoteNumberMethod 395
- EndnoteOptions 396
- EndnoteOptionsDlg 397
- EndnotePlacement 397
- EndnotePlacementDlg 398
- EndnoteSetNumberDlg 398
- ?EndnoteSpace 912
- EndnotesSpaceBetween 398
- ENDSWITCH 228
- ENDWHILE 237
- EnterKey 399
- EnvelopeDlg 399
- Environment preferences
  - beep 638
  - hyphenation 643
  - menu 651
  - setting default 642
- Equation
  - creating box, open editor 399
  - Equation Editor, toggling 400
  - palette, displaying 400
- Equation, viewing
  - next pane 609
  - previous pane 609
- EquationBaseFontDlg 399
- EquationCreate 399
- EquationPalette 400
- EquationPaletteToggle 400
- EquationRetrieveDlg 400
- Equations in boxes *see* BoxEquation
  - commands
- EquationZoomDlg 401
- ERROR 160
- Error condition
  - asserting 77
  - executing a Label statement 201
  - responding to 160
- Error value, returning in a variable 78
- ErrorNumber 11, 78
- Errors in macros 33

- Errors, uninitialized variables 236
- Esc, responding to a Cancel condition 90
- Examples, macro 48-69
- EXISTS 161
- Exit *see* Close
- ExpandDlg 401
- Expressions 7, 19
  - Arithmetic 21
  - Bitwise 23
  - Character 20
  - Logical 22
  - Measurement 19
  - Numeric 19
  - Relational 22
- Extension, default document on/off 963

## F

- Family, outline *see* Outline family
- ?FeatureBar 912
- Figure editor
  - Image Tools palette, closing 401
  - Image Tools palette, displaying 401
- FigurePaletteDlg 401
- FigurePaletteDlgEnd 401
- FigureRetrieveDlg 402
- File
  - attributes, modifying 402
  - copying source file to destination file 403
  - deleting 403
  - renaming/moving 405
  - retrieving 404, 406
  - saving 408
- File, specify location preferences for
  - various types
    - backup files 646
    - document files 646
    - graphics files 647
    - hyphenation files 648
    - macro files 648
    - printer files 649
    - spreadsheet files 650
    - template files 650
- FileChangeAttributes 402
- FileChangeDir 402
- FileCopy 403
- FileDelete 403
- FileInsert 404
- FileInsertDlg 405
- FileMove 405

- Filename box control, adding to a dialog
  - box 120
- FileNew 405
- FileOpen 406
- FileOpenDlg 407
- FileRemoveDirectory 408
- FileSave 408
- FileSaveAsDlg 410
- Fill
  - color of border, specifying 264
  - gradient, customizing 411
  - graphics boxes/tables, creating for 410
  - pattern set, specifying 412
- Fill style
  - copying 414
  - creating 415
  - deleting 415
  - editing 416
  - retrieving 418
  - saving 419
  - saving/cancelling after editing 417
  - user-created, naming 418
- ?Fill 913
- FillColors 410
- ?FillGradientSteps 913
- ?FillGradientType 914
- FillSetGradient 411
- FillSetWPG2 412
- FillStyleCopy 414
- FillStyleCreate 415
- FillStyleDelete 415
- FillStyleEdit 416
- FillStyleEnd 417
- FillStyleName 418
- FillStyleRetrieve 418
- FillStyleSave 419
- ?FillType 914
- ?FillWPG2Pattern 914
- Find *see also* Search
  - case, matching 688
  - insertion point after search string 548
  - insertion point before search string 548
  - matched text, selecting 548
  - specific number, limiting to 547
  - text between insertion point and matched text 547
  - text with attributes 548
  - text with selected font 549
  - text with selected font size 552
- Find and Replace *see* Replace
  - word/code/phrase
- FindDlg 419
- FindFiles 420
- FirstLineIndent 421
- ?FirstLineIndent 915
- ?FirstPrintLineNum 915
- Floating cell *see also* Table cell
  - commas in numbers, displaying 427
  - creating 422
  - currency symbols in 428
  - currency symbols, aligning 423
  - currency symbols, specifying 428
  - date format 424
  - digits after decimal point 425
  - editing 422
  - formula, specifying 423
  - move insertion point to beginning of 421
  - move insertion point to end of 422
  - negative numbers, displaying 426
  - number type, specifying 425
  - round numbers when calculating in 426
- Floating point 13
- FloatingCellBegin 421
- FloatingCellCreate 422
- FloatingCellEdit 422
- FloatingCellEnd 422
- FloatingCellFormula 423
- FloatingCellNumAlignCurrency 423
- FloatingCellNumberDateFormat 424
- FloatingCellNumberDecDigits 425
- FloatingCellNumberFormat 425
- FloatingCellNumberNegNumber 426
- FloatingCellNumberRound 426
- FloatingCellNumberUseCommas 427
- FloatingCellNumberUseCurrency 428
- FloatingCellNumSelectCurrency 428
- FlushRight 430
- Font 430
  - Bold, specifying 915
  - current size, showing value 918
  - current size, specifying 918
  - Double Underline, turning on/off 433, 916
  - Extra Large, turning on/off 433, 916
  - Fine, turning on/off 434, 916
  - Hidden, turning on/off 434
  - Italics, turning on/off 434, 916
  - Large, turning on/off 435, 917
  - Outline, turning on/off 435, 917
  - Redline, turning on/off 435, 917
  - relative size 251
  - Shadow, turning on/off 435, 917
  - size, ratios 251
  - size, setting 436
  - size, specifying for blocked text 250
  - Small Caps, turning on/off 436, 918
  - Small, turning on/off 436, 918
  - specifying 430
  - Strikeout, turning on/off 437, 918
  - Subscript, turning on/off 437, 918
  - Superscript, turning on/off 437, 919
  - Underline, turning on/off 438, 919
  - Very Large, turning on/off 438, 919
- Font, Power Bar
  - adding font size to list 655
  - adding font to list 654
- ?Font 915
- ?FontBold 915
- FontDlg 433
- ?FontDoubleUnderline 916
- FontDoubleUnderlineToggle 433
- ?FontExtraLarge 916
- FontExtraLargeToggle 433
- ?FontFine 916
- FontFineToggle 434
- FontHiddenToggle 434
- ?FontItalics 916
- FontItalicToggle 434
- ?FontLarge 917
- FontLargeToggle 435
- ?FontOutline 917
- FontOutlineToggle 435
- ?FontRedline 917
- FontRedlineToggle 435
- ?FontShadow 917
- FontShadowToggle 435
- FontSize 436
- ?FontSize 918
- ?FontSmall 918
- ?FontSmallCaps 918
- FontSmallCapsToggle 436
- FontSmallToggle 436
- ?FontStrikeout 918
- FontStrikeoutToggle 437
- ?FontSubscript 918
- FontSubscriptToggle 437
- ?FontSuperscript 919
- FontSuperscriptToggle 437
- ?FontUnderline 919
- FontUnderlineToggle 438
- ?FontVeryLarge 919
- FontVeryLargeToggle 438
- FooterA 438
- ?FooterAOccur 920
- FooterB 439

- ?FooterBOccur 920
  - Footers
    - blank space above, specifying 440
    - creating/editing/cancelling A 438
    - creating/editing/cancelling B 439
    - placement, specifying 468
  - FooterSeparationDistance 440
  - ?FooterSeparator 920
  - ?Footnote 920
  - ?FootnoteContinued 921
  - FootnoteContinuedMessage 440
  - FootnoteCreate 441
  - FootnoteEdit 441
  - FootnoteEditDlg 441
  - FootnoteMinimum 442
  - ?FootnoteMinimum 921
  - FootnoteNewNumber 442
  - FootnoteNewNumberDlg 443
  - FootnoteNumberDecrement 443
  - FootnoteNumberDisplay 443
  - FootnoteNumberIncrement 443
  - FootnoteOptions 444
  - FootnoteOptionsDlg 445
  - ?FootnotePosition 921
  - ?FootnoteRestartsPage 921
  - Footnotes *see also* Endnotes
    - creating 441
    - editing 441
    - numbering, decreasing by one 443
    - numbering, displaying current 443
    - numbering, increasing by one 443
    - numbering, specifying new 442
    - options, specifying 444
    - printing (continued...) message 440
    - separator line between text/footnotes, specifying 445
    - space between footer & text, specifying 447
    - text to keep together, specifying 442
  - FootnoteSeparatorLine 445
  - ?FootnoteSpace 922
  - FootnoteSpaceBetween 447
  - FOR 163
  - FOR-ENDFOR 27, 163
  - ForcePage 447
  - ForcePageDlg 448
  - FOREACH 164
  - FOREACH-ENDFOR 164
  - FormatMarginsDlg 448
  - Formulas *see also* Table, formulas in; Table, numbers in
    - calculating all in document 760
    - contents of current cell, converting to 821
    - copying to current cell 790
    - entering at document level 757
    - entering in table edit mode only 757
    - placing in cell 792
    - specifying which to insert in cell 799
    - Table Formula Control Bar, displaying 800
    - values, calculating 759
  - FORNEXT 166
  - FORNEXT-ENDFOR 166
  - FRACTION 167
  - Frame control, adding to a dialog box 121
  - FUNCTION 168
  - Functions 29
  - Functions, sharing between macro files 234
- G**
- Generate, document 448
  - GenerateDlg 448
  - GenerateDoc 448
  - GetAcceleratorText 449
  - GetData 450
  - GETNUMBER 171
  - GETSTRING 172
  - Getting Started 3
  - GETUNITS 173
  - GLOBAL 11, 174
  - Global arrays, creating 174
  - GO 28, 175
  - Grammatik, opening 862
  - GraphicBoxStyleDlg 452
  - GraphicCreateBox 452
  - GraphicCreateChart 452
  - GraphicCreateTextArt 453
  - GraphicCustomMenuDlg 453
  - Graphics
    - files, specifying location 647
    - import, specifying preferences 653
  - Graphics box *see also* Box commands
    - attaching to page/paragraph/character 278
    - sizing, specifying 454
  - Graphics line *see also* Line
    - counting in current document 924
    - creating 455
    - current number of in document, specifying 925
    - deleting 456
    - editing 456
    - editing next 457
    - editing previous 457
    - length of, modifying 459
    - length of, specifying 925
    - saving/cancelling changes to 458, 463
    - style, modifying 461
    - style, specifying 925
    - thickness, specifying 461, 926
  - Graphics line color
    - blue, specifying amount of 922
    - green, specifying amount of 923
    - predefined, using 463
    - red, specifying amount of 923
    - shading 923
    - specifying 454
    - using current/choosing another 927
  - Graphics line, position on page
    - column(s), aligning between 924
    - horizontal alignment, specifying 458, 924
    - horizontal or vertical, specifying 462, 927
    - horizontal position, specifying 925
    - spacing above, specifying 926
    - spacing below, specifying 922
    - vertical alignment, specifying 927
    - vertical position, specifying 464, 928
    - white space between line/text, specifying 460
  - GraphicsBoxCaptionDlg 453
  - GraphicsBoxContentDlg 453
  - GraphicsBoxFindDlg 454
  - GraphicsBoxSizeDlg 454
  - GraphicsFigureEditAllDlg 454
  - ?GraphicsLineBottomSpacing 922
  - GraphicsLineColor 454
  - ?GraphicsLineColorBlue 922
  - ?GraphicsLineColorGreen 923
  - ?GraphicsLineColorRed 923
  - ?GraphicsLineColorShading 923
  - ?GraphicsLineCount 924
  - GraphicsLineCreate 455
  - GraphicsLineDelete 456
  - GraphicsLineEdit 456
  - GraphicsLineEditNext 457
  - GraphicsLineEditPrevious 457
  - GraphicsLineEnd 458
  - ?GraphicsLineHorizontalAlign 924
  - ?GraphicsLineHorizontalColumn 924
  - GraphicsLineHorizontalPosition 458
  - ?GraphicsLineHorizontalPosition 925
  - GraphicsLineLength 459

- ?GraphicsLineLength 925
- ?GraphicsLineNum 925
- GraphicsLineSpacing 460
- GraphicsLineStyle 461
- ?GraphicsLineStyle 925
- GraphicsLineThickness 461
- ?GraphicsLineThickness 926
- ?GraphicsLineTopSpacing 926
- GraphicsLineType 462
- ?GraphicsLineType 927
- GraphicsLineUpdateDisplay 463
- GraphicsLineUseColor 463
- ?GraphicsLineUseColor 927
- ?GraphicsLineVerticalAlign 927
- GraphicsLineVerticalPosition 464
- ?GraphicsLineVerticalPosition 928
- GraphicsStylesDlg 464
- Gray scale of angle *see the*
  - BoxImageHalftoneAngle commands
- Grid lines in tables, turn on/off 801
- Group box control, adding to a dialog box 123

## H

- Halftone of image angle 888
- Handle, dialog, returning in a variable 152
- Hard Back Tab 252
- Hard Left Tab, insert 800
- Hard return, insert 466
- Hard Space, inserting 466
- HardColumnBreak 465
- HardPageBreak 465
- HardPageBreakInsert 465
- HardReturn 466
- HardReturnInsert 466
- HardSpace 466
- Header
  - blank space below, specifying 469
  - creating/editing/cancelling A 466
  - creating/editing/cancelling B 467
  - designating current row as 802
  - placement of, specifying 468
- Header example 37
- HeaderA 466
- ?HeaderAOccur 928
- HeaderB 467
- ?HeaderBOccur 928
- HeaderFooterDistanceDlg 468
- HeaderFooterDlg 468
- HeaderFooterPlacement 468
- HeaderFooterPlacementDlg 469

- HeaderSeparationDistance 469
- ?HeaderSeparator 929
- Help
  - contents, displaying 470
  - context-sensitive for active item, displaying 470
  - features, listing 471
  - instructions, displaying 471
  - keystrokes/templates, displaying information about 471
  - mouse/keystroke, using to get information 472
  - Tutorial, beginning 472
- HelpCoachDlg 470
- HelpContents 470
- HelpContextSensitive 470
- HelpHowDoI 471
- HelpHowToUseHelp 471
- HelpKeystrokes 471
- HelpMacroManual 471
- HelpSearchForHelpOn 472
- HelpTutorial 472
- HelpWhats 472
- HiddenText 473
- HiddenTextShowAll 473
- HideBars 474
- Highlight *see* Select
- HLineCreate 474
- Horizontal line control, adding to a dialog box 125
- Hot spot control, adding to a dialog box 126
- Hypertext 474
- HypertextControlBar 475
- HypertextCreate 475
- HypertextCreateDlg 476
- HypertextDelete 476
- HypertextEdit 476
- HypertextEditDlg 477
- HypertextExecute 477
- ?HypertextLast 929
- HypertextNext 478
- HypertextPrevious 478
- HypertextReturnFrom 478
- ?HypertextType 929
- Hyphen 479
- Hyphenation 479
  - activating 479
  - cancelling for particular word 479
  - files, specifying location 648
  - Hyphen Soft Return code, inserting 480
  - preferences, specifying 643

- zone, setting lower limit 480
- zone, setting upper limit 480
- ?Hyphenation 929
- HyphenationIgnore 479
- HyphenationSoftReturn 480
- HyphenationZoneLeft 480
- HyphenationZoneRight 480

## I

- Icon control, adding to a dialog box 127
- IF 176
- IF-ELSE-ENDIF 25, 176
- IFPLATFORM 178
- IFPLATFORM-ENDIFPLATFORM 26, 178
- Illustrations *see* Box image
- Image
  - halftone angle 1, specifying 888
  - halftone angle 2, specifying 888
  - halftone angle 3, specifying 889
  - halftone angle, specifying 888
  - halftone LPI, specifying 889
  - within graphics box, moving 310
- Import
  - conditions for NetWare SQL query, specifying 492
  - database fields, specifying 486
  - DataPerfect panel to retrieve, specifying 488
  - field conditions for query, specifying 489
  - field names as heading, using 488
  - file to import, specifying 487
  - format, specifying 485, 491
  - NetWare SQL database user ID, specifying 492
  - NetWare SQL dictionary path, specifying 486
  - NetWare SQL file path, specifying 487
  - password to import NetWare SQL, specifying 489
  - range of cells, specifying 490
  - server, specifying 491
  - spreadsheet/database data into current document 482
- Import preferences
  - graphics format 653
  - merge delimiters, specifying 644
- ImportCreateDlg 481
- ImportDlg 481
- ImportDoEdit 481

- ImportDoImport 482
  - ImportDoLink 482
  - ImportDoUpdate 482
  - ImportEditDlg 483
  - ImportOptionsDlg 483
  - ImportSetAsciiEncap 483
  - ImportSetAsciiFieldDelimiter 484
  - ImportSetAsciiRecordDelimiter 484
  - ImportSetAsciiStrip 485
  - ImportSetDestination 485
  - ImportSetDictPath 486
  - ImportSetFields 486
  - ImportSetFileName 487
  - ImportSetFilePath 487
  - ImportSetIncludeNames 488
  - ImportSetPanel 488
  - ImportSetPassword 489
  - ImportSetQuery 489
  - ImportSetRange 490
  - ImportSetServer 491
  - ImportSetSource 491
  - ImportSetSQLQuery 492
  - ImportSetUserID 492
  - ImportUpdateDlg 493
  - Indent 493
  - Indent right 493
  - ?Indent 930
  - Index
    - format, specifying 494
    - generating 448
    - reference to heading and subheading, creating 543
    - text, marking 496, 543
  - IndexDefineDlg 493
  - IndexDefinition 494
  - ?IndexDefinition 930
  - IndexMark 496
  - INDIRECT 11, 179
  - Information exchange *see* DDE commands; OLE commands
  - InhibitInput 496
  - InhibitInput(Off), using with SENDKEYS WAIT 216
  - InitialCodesStyleDlg 497
  - ?Initials 930
  - ?InOutline 930
  - Insert
    - bullet 497
    - filename and path 498
    - mode, turning on/off 850
    - number 497
  - InsertDefaultBullet 497
  - InsertFilename 497
  - InsertFilenameWithPath 498
  - Insertion point, movement in columns
    - bottom 621
    - first 622
    - last 622
    - previous 623
    - top 623
  - Insertion point, movement in tables
    - beginning of row 635
    - beginning of selection 631
    - bottom of cell 632
    - bottom of column 633
    - cell 631
    - down one cell 632
    - down one row 619
    - end of row 635
    - first cell 631
    - last cell 634
    - left one character 620
    - next cell 632, 635
    - previous cell 633, 636
    - previous position 634
    - right one character 619
    - top of cell 633
    - top of column 634
    - up one cell 633
    - up one row 620
  - Insertion point, movement in text
    - after specified character 620
    - beginning of current line 625
    - beginning of line (before codes) 626
    - bottom of current document 623
    - bottom of current page 627
    - bottom of selection 630
    - down one line 625
    - down one screen 629
    - end of current line 625
    - end of line (after codes) 626
    - left edge of screen 629
    - next occurrence of specified code 624
    - next page 627
    - next paragraph 628
    - next word 636
    - one character to left 621
    - one character to right 621
    - previous page 628
    - previous paragraph 629
    - previous position 624
    - previous word 636
    - right edge of screen 629
    - specified page number 626
    - top of current document 623
    - top of current page 628
  - top of current selection 630
  - top of document (before codes) 624
  - up one line 626
  - up one screen 630
  - Insertion point, position, in outline or not 930
  - InsertOtherCodesDlg 498
  - INTEGER 181
  - ?InterimVersion 931
  - InvokeDialog 498
- ## J
- Jump *see* Hypertext
  - Justification 499
  - ?Justification 931
  - JustifyAll 499
  - JustifyCenter 500
  - JustifyFull 500
  - JustifyLeft 500
  - JustifyRight 500
- ## K
- KeepTextTogetherDlg 501
  - Kerning 501
  - ?Kerning 931
  - KerningSpacing 502
  - Keyboard menu, resetting 686
  - Keyboard, selecting 502
  - KeyboardSelect 502
  - KeyboardSelectDlg 503
  - Keystring 985
  - Keystroke Macros 69
  - Keystrokes, recording 5
  - Keystrokes, sending to the current application 216
- ## L
- LABEL 182
    - calling 92, 93, 199, 200, 202, 204
  - Label statement
    - Cancel condition 198, 199
    - Error condition 201, 202
    - Not Found condition 203, 204
  - Labels 29
    - calling 175
    - creating names with INDIRECT 179
    - customized definition, creating 503
    - definition, selecting 507
    - mode, turning off 507
  - Labels file
    - creating 505

- description, adding to existing file
  - 506
  - selecting 506
- LabelsDefine 503
- LabelsDelete 504
- LabelsDlg 505
- LabelsFileCreate 505
- LabelsFileDescription 506
- LabelsFileSelect 506
- LabelsOff 507
- LabelsSelect 507
- ?LabelsToDisplay 931
- Language 507
- ?Language 932
- LanguageDlg 508
- LastOpenedAdd 508
- Layout, paper size
  - adding 609
  - deleting 611
  - editing 612
  - saving 612
  - selecting 614
- Layout, QuickFormat
  - specifying format 674
  - toggling 675
- Leading 508
- ?LeftChar 932
- ?LeftCode 932
- ?LeftHyphZone 932
- ?LeftMarginAdj 933
- Library, macro, sharing functions and procedures 234
- Library, macro, sharing functions and procedures 31
- Line *see also* Graphics line; Table, lines in
  - attributes, copying 521
  - horizontal, creating 474
  - vertical spacing, specifying 521
  - vertical, creating at left margin 855
- Line height, specifying 509
- Line numbering
  - beginning number, specifying 514
  - method, specifying 513
  - turning on/off 511
- Line segment
  - blue, specifying amount of 935
  - color, specifying 514
  - counting 936
  - creating 515
  - custom line pattern sequence, ending 518
  - dash/space length in custom line pattern 518
  - dash/space length in custom line segment 517
  - deleting 516
  - distance between, specifying 520
  - editing 516
  - green, specifying amount of 936
  - red, specifying amount of 936
  - series of commands, ending 517
  - shading, specifying 937
  - spacing, specifying 938
  - user data, setting 937
  - width, specifying 520, 938
  - WPG2 pattern, setting 519, 938
- Line style
  - creating 522
  - deleting 523
  - editing 524
  - ending 525
  - library, retrieving to 526
  - library, saving 526
  - renaming 525
- ?Line 933
- LineCreateDlg 509
- LineEditDlg 509
- LineHeight 509
- ?LineHeight 933
- LineHeightDlg 510
- LineHyphenationDlg 510
- ?LineNumber 933
- LineNumbering 511
- ?LineNumbering 933
- LineNumberingDlg 513
- ?LineNumberingInterval 934
- LineNumberingMethod 513
- ?LineNumberMethod 934
- LineNumberSet 514
- ?LineNumCountBlankLines 934
- ?LineNumNewspaperCols 935
- ?LineNumPosition 935
- Lines in tables *see* Tables, lines in
- ?LineSegmentBlue 935
- LineSegmentColor 514
- ?LineSegmentCount 936
- LineSegmentCreate 515
- LineSegmentDelete 516
- LineSegmentEdit 516
- LineSegmentEnd 517
- ?LineSegmentGreen 936
- ?LineSegmentRed 936
- LineSegmentSetUser 517
- LineSegmentSetUserData 518
- ?LineSegmentSetUserData 937
- LineSegmentSetUserEnd 518
- LineSegmentSetWPG2 519
- ?LineSegmentShading 937
- LineSegmentSpacing 520
- ?LineSegmentSpacing 938
- LineSegmentWidth 520
- ?LineSegmentWidth 938
- ?LineSegmentWPG2Pattern 938
- LineSpacing 521
- ?LineSpacing 939
- LineSpacingDlg 521
- LineStyleCopy 521
- LineStyleCreate 522
- LineStyleDelete 523
- LineStyleEdit 524
- LineStyleEnd 525
- LineStyleName 525
- LineStyleRetrieve 526
- LineStyleSave 526
- Link *see also* the Hypertext commands
  - between server and client applications, accepting 568
  - between server and client applications, cancelling 569
  - creating/performing 482
  - file used in, changing 569
  - linked object, cancelling 568
  - modifying/performing 481
  - pasting 616
  - revising 482
  - server application, reinitializing 569
  - updating 527, 571
  - updating object, specifying how to 571
- LinkUpdate 527
- List
  - generating 448
  - item, marking 544
  - reference, creating 544
- List box control, adding to a dialog box 129
- List definition
  - creating 530
  - Define List dialog box, deleting in 530
  - graphics box, changing 531
  - numbering format, changing 531
  - page numbering format, editing 535
  - page, defining 527
  - renaming 532
  - retrieving 532
  - style, editing 533



- List item, adding to combo box/list box/popup box 131
  - ListDefineDlg 527
  - ListDefinition 527
  - ?ListDefinition 939
  - ListDefinitionCreate 530
  - ListDefinitionDelete 530
  - ListDefinitionEditBox 531
  - ListDefinitionMode 531
  - ListDefinitionRename 532
  - ListDefinitionRetrieve 532
  - ListDefinitionStyle 533
  - ListEditPageFormat 535
  - ListMark 535
  - LOCAL 11, 183
  - Local arrays, creating 104, 183
  - Location of files *see* Files, specify location preferences for various
  - Lock/unlock cells in tables 767, 797
  - Logical Expressions 22
  - Logical Operators 15
  - Loop
    - advancing to next iteration 194
    - ending 88
    - FOR-ENDFOR 163
    - FOREACH-ENDFOR 164
    - FORNEXT-ENDFOR 166
    - REPEAT-UNTIL 212
    - WHILE-ENDWHILE 237
  - Loop statements 8, 26
  - Lowercase, turn uppercase 336
- M**
- Macro
    - adding to macro submenu 537
    - assigning to a Button Bar 69
    - assigning to a keyboard 69
    - assigning to a menu bar 69
    - CALENDAR.WCM 66
    - CALLBACK.WCM 54
    - CB\_SCBAR.WCM 58
    - command-based language 5
    - compiling 5
    - control statements 25
    - deleting from menu 538
    - editing 33
    - ending 211, 213
    - errors 33
    - Feature Bar, displaying 536
    - files, specifying location 648
    - hints 4
    - information in Help, displaying 471
    - keyboard/menu input during execution of 496
    - languages 5
    - pausing 205, 236, 538
    - pausing at specified command 617
    - pausing until specific key pressed 618
    - playing 32
    - prompting for user input 210
    - purpose of 6
    - quitting 211
    - recording 32
    - resuming after pause 618
    - sharing functions and procedures 234
    - slowing execution 219
    - starting from another macro 94
    - starting nested macros 215
    - status prompt line, displaying 538
    - using an application in 76
    - VIEWBARS.WCM 37
  - Macro Control Statements 25
  - Macro examples 48
  - Macro Facility 4
    - compiling macros 5
  - Macro File Library 31
    - returning from 213
  - Macro Language Concepts 6
  - Macro library, sharing functions and procedures 234
  - MacroAssignDlg 536
  - MacroCommandInserter 536
  - MacroDialogResult 11, 126, 134, 153
  - MacroEditControlBar 536
  - ?MacroExecute 939
  - MacroMenuAdd 537
  - MacroMenuDelete 538
  - Macro menus 50
  - MacroPause 538
  - ?MacroPlaying 940
  - ?MacroRecord 940
  - MacroStatusPrompt 538
  - MailDlg 539
  - ?MajorVersion 940
  - Margin
    - bottom, setting 540
    - left, adjusting 539
    - left, setting 541
    - releasing 541
    - right, adjusting 540
    - right, setting 541
    - top, setting 542
  - MarginAdjustLeft 539
  - MarginAdjustRight 540
  - MarginBottom 540
  - ?MarginBottom 940
  - MarginLeft 541
  - ?MarginLeft 941
  - MarginReleaseInsert 541
  - MarginRight 541
  - ?MarginRight 941
  - MarginTop 542
  - ?MarginTop 941
  - MarkCrossRef 542
  - MarkCrossRefUpdateList 542
  - MarkIndex 543
  - MarkIndexUpdateList 543
  - MarkList 544
  - MarkListUpdateList 544
  - MarkTableOfAuthorities 544
  - MarkTableOfContents 545
  - Marking text *see* the Bookmark commands; the Mark commands
  - MarkToAUpdateList 545
  - Master Document *see also* Subdocument generating 448
  - MasterDocCondense 545
  - MasterDocCondenseDlg 546
  - MasterDocExpand 546
  - MasterDocSubdocDlg 547
  - MatchExtendSelection 547
  - MatchLimit 547
  - MatchPositionAfter 548
  - MatchPositionBefore 548
  - MatchSelection 548
  - MatchWithAttributes 548
  - MatchWithFont 549
  - MatchWithFontSize 552
  - Measure, unit of, setting default 106
  - Measure, units of, set default 664
  - Measurement Expressions 19
  - MENU 184
    - display, setting default 651
    - environment preferences, specifying 651
  - MenuItemAdd 552
  - MenuItemDelete 554
  - MENULIST 185
  - Menus, macro 50
  - MenuSelect 554
  - MenuSelectDlg 555
  - Merge
    - blank fields, removing 555
    - condition, defining 559
    - data file, specifying 560
    - delimiters, specifying 644
    - envelope, creating 560
    - form and data files, associating 562

- number of times for each record,
    - setting 564
  - performing 565
  - records to include, marking 563
  - records to merge, identifying 566
  - sending file to printer 564
  - separating documents with page break 563
  - type of file to merge, specifying 562
  - Merge codes
    - displaying 652
    - hiding/showing/displaying as icons in merged document 558
    - hiding/showing/displaying as icons in range 558
    - inserting 556
  - MergeBlankField 555
  - MergeCode 556
  - MergeCodesDisplayEdit 558
  - MergeCodesDisplayRun 558
  - MergeCodesDlg 559
  - MergeCondition 559
  - MergeDataFile 560
  - MergeDlg 560
  - ?MergeDone 941
  - MergeEnvelope 560
  - MergeEnvelopeDefinitionBegin 561
  - MergeEnvelopeDefinitionEnd 561
  - MergeFileAssociate 562
  - MergeFilesDlg 562
  - MergeFileType 562
  - MergeFormDlg 563
  - MergeMark 563
  - MergePageBreak 563
  - MergePrintText 564
  - MergeRepeat 564
  - MergeRun 565
  - MergeSelect 566
  - MergeTableDataDlg 566
  - MergeTextDataDlg 566
  - Message box, creating 186
  - MESSAGEBOX 44, 186
  - ?MinorVersion 941
  - MinusSign 567
  - MMPLAY 189
  - MMSPEAK 190
  - MMSPEAKCLIPBOARD 191
  - MMSTOPSPEECH 192
  - MOD 13
  - Modal dialog boxes 42
  - Modeless Dialog Boxes 42
  - Mouse actions, recording 5
  - Mouse String 987
  - MoveModeEnd 567
  - Moving the insertion point *see* Insertion point movement
  - Music *see* Sound clip
- N**
- ?Name 942
  - Named regions 969
  - Narration *see* Sound clip
  - NEST 193
  - Nested macros, starting 215
  - NetWare SQL file path, specifying 487
  - ?Network 942
  - NEWDEFAULT 193
  - NewHFWOccurrence 567
  - NEXT 194
  - NOT 12
  - Not Found condition
    - asserting 77
    - responding to 195
  - NOTFOUND 195
  - ?NotFound 942
  - NTOC 196
  - Numbering items in document *see the* Counter commands
  - Numbering style, in box captions 283
  - Numbering, page *see* Page number
  - NumberingOptionDlg 568
  - Numbers *see also* Tables, numbers in; Formulas
    - converting number strings to measurements 224
    - converting to a character string 197
    - converting to character equivalents 196
    - creating from a string 222
    - entering in an edit box 171
  - Numeric Expressions 19
  - NUMSTR 197
- O**
- Object files 5
  - Object Linking & Embedding *see the* OLE commands
  - Offset of binding, specifying 255
  - OLE link, pasting 616
  - OLEAcceptLinks 568
  - OLECancelLink 568
  - OLECancelLinks 569
  - OLEChangeLink 569
  - OLEInitializeLinks 569
  - OLEInsertDlg 570
  - OLEInsertObject 570
  - OLELinksDlg 570
  - OLEPasteDlg 571
  - OLESetLinkUpdate 571
  - OLEUpdateLink 571
  - ONCANCEL 44, 198
  - ONCANCEL CALL 199
  - ONDDEADVISE CALL 200
  - ONERROR 44, 201
  - ONERROR CALL 202
  - ONNOTFOUND 44, 203
  - ONNOTFOUND CALL 204
  - Operator Precedence 18
  - Operators
    - Arithmetic 13
    - Binary 13
    - Bitwise 16
    - Logical 15
    - Precedence 18
    - Relational 14
    - Unary 12
  - Original Document Backup, turning on/off 253
  - Outline
    - body text, changing to 575
    - body text, hiding or showing 574
    - feature bar, displaying 854
    - feature bar, turning off 581
    - feature bar, turning on 572
    - hiding outline and showing body text 581
    - inserting hard return in 466
    - level, changing 615
    - lowest level to show, specifying 585
    - new in specified style, beginning 572
    - next paragraph, blocking 575
    - next, blocking 585
    - text between outline and body text, toggling 575
    - turning off 576
  - Outline family
    - copying and adjusting 576
    - current, hiding 578
    - deleting and adjusting 576
    - dragging down 577
    - dragging up 578
    - Edit in Outline Mode 579
    - hidden, displaying 580
    - insertion point, moving to beginning of previous 581
    - insertion point, moving to end of next 577

- insertion point, moving to next paragraph 579
  - insertion point, moving to previous paragraph 578
  - moving 579
  - next level, adjusting to 577
  - selecting 580
  - Outline of font character 435
  - Outline paragraph
    - copying 582
    - deleting 582
    - dragging down 583
    - dragging up 583
    - level, specifying 584
    - moving 583
    - next level, adjusting to 582
    - number, setting 585
    - previous level, adjusting to 584
    - previous, blocking 600
    - selecting 584
  - Outline style
    - and level, specifying 594
    - copying 586
    - creating 588
    - deleting 589
    - describing 591
    - numbering format, specifying 595
    - retrieving 597
    - saving 598
    - selecting 599
    - user-created, renaming 597
  - OutlineBarSetup 572
  - OutlineBeginNew 572
  - OutlineBodyText 574
  - OutlineBodyTextToggle 575
  - OutlineChangeToBodyText 575
  - OutlineDefineDlg 575
  - OutlineDown 575
  - OutlineEnd 576
  - OutlineFamilyCopy 576
  - OutlineFamilyCut 576
  - OutlineFamilyDemote 577
  - OutlineFamilyDown 577
  - OutlineFamilyDragDown 577
  - OutlineFamilyDragUp 578
  - OutlineFamilyHide 578
  - OutlineFamilyLeft 578
  - OutlineFamilyMove 579
  - OutlineFamilyPromote 579
  - OutlineFamilyRight 579
  - OutlineFamilySelect 580
  - OutlineFamilyShow 580
  - OutlineFamilyUp 581
  - OutlineHideShowingBodyText 581
  - OutlineNumberDlg 581
  - OutlineOff 581
  - OutlineParagraphCopy 582
  - OutlineParagraphDelete 582
  - OutlineParagraphDemote 582
  - OutlineParagraphDragDown 583
  - OutlineParagraphDragUp 583
  - OutlineParagraphMove 583
  - OutlineParagraphNumber 584
  - OutlineParagraphPromote 584
  - OutlineParagraphSelect 584
  - OutlineRight 585
  - OutlineSetParagraphNumber 585
  - OutlineShow 585
  - OutlineStyleCopy 586
  - OutlineStyleCreate 588
  - OutlineStyleDelete 589
  - OutlineStyleDescription 591
  - OutlineStyleEditBegin 591
  - OutlineStyleEditEnd 593
  - OutlineStyleLevel 594
  - OutlineStyleNumberFormat 595
  - OutlineStyleNumbersOnly 596
  - OutlineStyleRename 597
  - OutlineStyleRetrieve 597
  - OutlineStyleSave 598
  - OutlineStyleSelect 599
  - OutlineUp 600
  - Output, Jogger, selecting 601
  - OutputOption 601
  - OutputOptionsSetOffsetJogger 601
  - ?Overstrike 942
  - OverstrikeCreate 602
  - OverstrikeCreateDlg 602
  - OverstrikeEdit 603
  - Overview
    - Coach 969
- P**
- Page break, insert hard 465
  - Page number
    - decreasing 604
    - displaying 604
    - forcing odd/even 447
    - format, displaying 604
    - formatting 605
    - increasing 605
    - method, specifying 606
    - new, specifying 603
    - position & appearance on page, specifying 606
  - settings 608
  - ?Page 942
  - ?PageBorder 943
  - PageBorderDlg 603
  - PageNumber 603
  - PageNumberDecrement 604
  - PageNumberDisplay 604
  - PageNumberDisplayFormat 604
  - PageNumberFormat 605
  - ?PageNumberFormat 943
  - PageNumberIncrement 605
  - PageNumberingDlg 605
  - ?PageNumberingOn 943
  - ?PageNumberingValue 943
  - PageNumberMethod 606
  - ?PageNumberMethod 944
  - PageNumberPosition 606
  - ?PageNumberPosition 944
  - PageNumberSettingsDlg 608
  - ?PageSuppress 944
  - PageSuppressDlg 608
  - PageZoomFull 609
  - Palette *see* Color Palette; the Color commands
  - PaneNext 609
  - PanePrevious 609
  - ?PaperLength 945
  - PaperSizeAdd 609
  - PaperSizeDelete 611
  - PaperSizeDlg 612
  - PaperSizeResave 612
  - PaperSizeSelect 614
  - ?PaperSizeType 945
  - ?PaperWidth 945
  - ?ParagraphBorder 945
  - ParagraphBorderDlg 614
  - ParagraphFormatDlg 615
  - ParagraphHangingIndent 615
  - ?ParagraphNumber 946
  - ParagraphNumberDefDlg 615
  - ?ParagraphNumberMethod 946
  - ParagraphSpacing 616
  - Parameters 8
  - Paste format, specifying 617
  - PasteLink 616
  - PasteSpecial 617
  - Path
    - backup files 946
    - current directory 947
    - database files 947
    - database files, supplemental 947
    - document files 947
    - graphics files 947

- graphics files, supplemental 947
- hyphenation files 948
- labels files 948
- macros files 948
- macros files, supplemental 948
- printer files 948
- printer files, supplemental 948
- program files 949
- shared code files 949
- spreadsheet files 949
- spreadsheet files, supplemental 949
- template files 949
- template files, supplemental 949
- WPWIN60.EXE 949
- ?Path 946
- ?PathBackup 946
- ?PathCurrent 947
- ?PathDatabase 947
- ?PathDatabaseSupplemental 947
- ?PathDocument 947
- ?PathGraphics 947
- ?PathGraphicsSupplemental 947
- ?PathHyphenation 948
- ?PathLabels 948
- ?PathMacros 948
- ?PathMacrosSupplemental 948
- ?PathPrinters 948
- ?PathPrintersSupplemental 948
- ?PathProgram 949
- ?PathShared 949
- ?PathSpreadsheet 949
- ?PathSpreadsheetSupplemental 949
- ?PathTemplate 949
- ?PathTemplateSupplemental 949
- PAUSE 205
- PauseCommand 617
- PauseKey 618
- ?PauseKey 950
- PauseSet 618
- PERSIST 11, 206
- Persist arrays, creating 206, 207
- PERSISTALL 207
- Platforms, compiling for different 178
- Playing macros 5, 32
- Popup button control, adding to a dialog box 132
- Port, current printer 907
- PosCellDown 619
- PosCellNext 619
- PosCellPrevious 620
- PosCellUp 620
- PosCharacter 620
- PosCharNext 621
- PosCharPrevious 621
- PosColBottom 621
- PosColFirst 622
- PosColLast 622
- PosColNext 622
- PosColPrevious 623
- PosColTop 623
- PosDocBottom 623
- PosDocTop 623
- PosDocVeryTop 624
- PosFunction 624
- PosGoPrevious 624
- PosGoToDlg 625
- ?Position 950
- Positioning graphics boxes *see* Box, position on page
- Positioning graphics lines *see* Graphics line, position on page
- Positioning the insertion point *see* Insertion point movement; the Pos commands
- PosLineBeg 625
- PosLineDown 625
- PosLineEnd 625
- PosLineUp 626
- PosLineVeryEnd 626
- PosLineVeryEnd 626
- PosPage 626
- PosPageBottom 627
- PosPageNext 627
- PosPagePrevious 628
- PosPageTop 628
- PosParagraphNext 628
- PosParagraphPrevious 629
- PosScreenDown 629
- PosScreenLeft 629
- PosScreenRight 629
- PosScreenUp 630
- PosSelectBottom 630
- PosSelectTop 630
- PosTableBegin 631
- PosTableBlockBeg 631
- PosTableCell 631
- PosTableCellBottom 632
- PosTableCellDown 632
- PosTableCellNext 632
- PosTableCellPrevious 633
- PosTableCellTop 633
- PosTableCellUp 633
- PosTableColBottom 633
- PosTableColumnTop 634
- PosTableEnd 634
- PosTableGoPrevious 634
- PosTableRowBegin 635
- PosTableRowEnd 635
- PosTableTopCellNext 635
- PosTableTopCellPrevious 636
- POSTNET Barcode *see* the Barcode commands
- PosWordNext 636
- PosWordPrevious 636
- Power Bar, displaying 636
- Power Bar preferences
  - buttons, specifying 653
  - editing 654
  - font size, specifying 655
  - font, adding 654
- ?PowerBarActive 950
- PowerBarShow 636
- Precedence, operator 18
- PrefBackupDlg 637
- PrefBackupPath 637
- PrefBeep 638
- PrefButtonBar 639
- PrefDisplayDlg 639
- PrefDisplaySet 639
- PrefDocSummary 641
- PrefDocSummaryDlg 641
- PrefEnvironment 642
- PrefEnvSettingsDlg 643
- Preferences
  - document summary, setting default 641
  - enlarging/reducing size of text and graphics 665
  - features, specifying display 639
  - merge codes, displaying 652
  - page formatting codes, displaying 645
  - Reveal Code settings, displaying 658
  - Ruler Bar settings, specifying 660
  - saving 661
  - Table of Authorities format, specifying 664
  - units of measure settings, specifying 664
- Preferences, environment
  - beep 638
  - hyphenation 643
  - menu display 651
  - set default 642
- Preferences, file
  - backup files 646
  - backup path 637
  - documents 646
  - graphics files 647
  - hyphenation files 648

- macro files 648
  - printer files 649
  - spreadsheet files 650
  - template files 650
  - Preferences, import
    - graphics, specifying format 653
    - merge delimiter, specifying 644
  - Preferences, Power Bar
    - buttons, specifying 653
    - editing 654
    - font size, adding to list 655
    - font, adding to list 654
  - Preferences, print
    - default, setting 656
    - font size ratio, specifying 657
    - number of copies, specifying 655
  - Preferences, Status Bar
    - default, setting 661
    - options, specifying 662
  - PrefHyphenation 643
  - PrefImportDlg 644
  - PrefImportOptions 644
  - PrefLiteCodes 645
  - PrefLocationDlg 646
  - PrefLocationOfDocuments 646
  - PrefLocationOfGraphicsFiles 647
  - PrefLocationOfHyphenationFiles 648
  - PrefLocationOfMacroFiles 648
  - PrefLocationOfPrinterFiles 649
  - PrefLocationOfSpreadsheetFiles 650
  - PrefLocationOfTemplates 650
  - PrefMenu 651
  - PrefMergeCodes 652
  - PrefMetafileOptions 653
  - PrefPowerBar 653
  - PrefPowerBarEdit 654
  - PrefPowerBarFontFaceList 654
  - PrefPowerBarFontSizeList 655
  - PrefPrintCopies 655
  - PrefPrintDoc 656
  - PrefPrintRedline 656
  - PrefPrintSettingsDlg 657
  - PrefPrintSizeRatio 657
  - PrefRevealCodes 658
  - PrefRevealCodesDlg 659
  - PrefRuler 660
  - PrefSave 661
  - PrefStatusBar 661
  - PrefStatusBarEdit 662
  - PrefStatusBarOptions 662
  - PrefStatusBarOptionsDlg 663
  - PrefToA 664
  - PrefUnitsOfMeasure 664
  - PrefZoom 665
  - Print
    - color, specifying 671
    - document on disk 667
    - document, full 670
    - multiple copies 666
    - multiple pages 672
    - number of copies, specifying 666
    - output options, specifying 601
    - quality of graphics, specifying 671
    - quality of text, specifying 673
    - Redline preferences, specifying 656
    - selected text 673
    - size ratio, specifying 657
  - Print preferences
    - default, specifying 656
    - font size ratio, specifying 657
    - number of copies, specifying 655
  - PrintCopies 666
  - PrintCopiesBy 666
  - PrintDlg 667
  - PrintDocOnDisk 667
  - Printer
    - location of files, specifying 649
    - printer command, inserting 668
    - soft fonts, downloading 669
  - Printer port, current 907
  - PrinterCommand 668
  - PrinterCommandDlg 669
  - PrinterControlDlg 669
  - PrinterInitialize 669
  - PrinterPause 669
  - PrinterSelect 670
  - PrinterSelectDlg 670
  - PrintFullDoc 670
  - PrintGraphicsQuality 671
  - PrintInkColor 671
  - PrintMultiplePages 672
  - PrintPage 673
  - PrintSelection 673
  - PrintTextQuality 673
  - PROCEDURE 208
  - Procedures 30
  - Procedures, sharing between macro files 234
  - Product commands 9
  - Programming commands 9
  - PROMPT 210
  - Prompts, creating 210
  - Push button control, adding to a dialog box 134
- Q**
- QuickFinderDlg 674
  - QuickFormat 674
  - QuickFormatDlg 674
  - QuickFormatToggle 675
  - QuickmarkFind 675
  - QuickmarkSet 675
  - QUIT 211
- R**
- Radio button control, adding to a dialog box 136
  - Radix Base Conversions 20
  - ?Random 950
  - Range of colors, selecting *see* the Color commands
  - Real number
    - returning the fractional portion 167
    - returning the integer portion 181
  - Record a Macro 32
  - Redisplay 676
  - Redline *see also* Font
    - character, specifying 676
    - method, specifying 676
    - preferences, specifying 656
  - RedlineCharacter 676
  - ?RedlineCharacter 950
  - RedlineMethod 676
  - ?RedlineMethod 951
  - Relational Expressions 22
  - Relational Operators 14
  - REPEAT 212
    - default, setting 677
    - number of times, specifying 678
  - REPEAT-UNTIL 27, 212
  - RepeatDlg 677
  - RepeatSet 677
  - RepeatValue 678
  - Replace word/code/phrase
    - all instances 678
    - current 679
    - to beginning of document 679
    - to end of document 680
    - with specific font size 685
    - with specific font/attributes 683
    - with specific text/attributes 681
    - with specific text/codes 681
    - with uppercase/mixed case 682
  - ReplaceAll 678
  - ReplaceBackward 679
  - ReplaceCurrent 679
  - ReplaceDlg 680

- ReplaceForward 680
- ReplaceString 681
- ReplaceWithAttributes 681
- ReplaceWithCase 682
- ReplaceWithFont 683
- ReplaceWithFontSize 685
- ReselectLastSelection 686
- ResetKeyboardMenu 686
- RETURN 213
- Return statement
  - CALL 89
  - CASE 93
  - ONCANCEL CALL 199
  - ONDDEADWISE CALL 200
  - ONERROR CALL 202
  - ONNOTFOUND CALL 204
  - RETURN 213
- Reveal Codes
  - default settings, specifying 658
  - displaying 687
- RevealCodes 687
- ?RevealCodesActive 951
- ?RightChar 951
- ?RightCode 951
- ?RightHyphZone 952
- ?RightMarginAdj 952
- ?Row 952
- ?RowHeader 952
- ?RowHeight 952
- Rows, table *see* Table row
- Ruler Bar
  - default, setting 660
  - displaying 687
  - hiding 474
- ?RulerBarActive 953
- RulerBarShow 687
- RUN 215

**S**

- Save
  - all unsaved documents 688
  - preferences, specifying 661
  - selected text 705
- SaveAll 688
- ?SaveWorkspace 953
- Scroll bar 688
- Scroll bar control, adding to a dialog box 137
- Scroll bar, hide 474
- ScrollScreenLeft 688
- ScrollScreenRight 688
- Search *see also* Find
  - backward 690
  - entire document 691
  - forward 690
    - matching case 688
    - selected text 689
    - specific string of text 691
    - whole words only 689
  - Search and Replace *see* Replace
    - word/code/phrase
  - SearchCaseSensitive 688
  - SearchFindWholeWordsOnly 689
  - SearchInSelection 689
  - SearchNext 690
  - SearchPrevious 690
  - SearchString 691
  - SearchWrap 691
  - SecondaryPageNumber 692
  - ?SecondaryPageNumber 953
  - SecondaryPageNumberDecrement 692
  - SecondaryPageNumberDisplay 693
  - SecondaryPageNumberIncrement 693
  - SecondaryPageNumberMethod 693
  - ?SecondaryPageNumberMethod 953
  - Select
    - character, next 697
    - character, previous 697
    - column(s) 697, 698
    - document, to bottom of 699
    - document, to top of 699
    - document, to very top of 700
    - line(s) 700-702
    - menu 554
    - page 703
    - page, next 703
    - page, previous 704
    - paragraph 704
    - paragraph, next 704
    - paragraph, previous 705
    - previous selection 686
    - rectangle 705
    - screen down 706
    - screen left 706
    - screen right 707
    - screen up 707
    - sentence 707
    - sentence, next 708
    - sentence, previous 708
    - table 708
    - table cell(s) 694-696
    - table column 709
    - table row 710
    - table, from current cell to left 709
    - table, from current cell to right 709
    - tabular column 710
    - text and graphics 694
    - word 710
      - word, next 711
      - word, previous 711
  - Select functions
    - Select Mode, turning on/off 702, 710
    - selected text, deleting 699
    - selected text, saving as new file 705
  - SelectAll 694
  - SelectCell 694
  - SelectCellDown 694
  - SelectCellDownArrow 695
  - SelectCellLeft 695
  - SelectCellRight 696
  - SelectCellUp 696
  - SelectCellUpArrow 696
  - SelectCharNext 697
  - SelectCharPrevious 697
  - SelectColumnBottom 697
  - SelectColumnNext 698
  - SelectColumnPrevious 698
  - SelectColumnTop 698
  - SelectDelete 699
  - SelectDocBottom 699
  - SelectDocTop 699
  - SelectDocVeryTop 700
  - ?SelectedText 954
  - ?SelectedTextSize 954
  - SelectLineBegin 700
  - SelectLineDown 700
  - SelectLineEnd 701
  - SelectLineUp 701
  - SelectLineVeryBegin 701
  - SelectLineVeryEnd 702
  - SelectMode 702
  - SelectOff 702
  - SelectOn 703
  - SelectPage 703
  - SelectPageNext 703
  - SelectPagePrevious 704
  - SelectParagraph 704
  - SelectParagraphNext 704
  - SelectParagraphPrevious 705
  - SelectRectangle 705
  - SelectSave 705
  - SelectScreenDown 706
  - SelectScreenLeft 706
  - SelectScreenRight 707
  - SelectScreenUp 707
  - SelectSentence 707
  - SelectSentenceNext 708
  - SelectSentencePrevious 708

- SelectTable 708
- SelectTableColumn 709
- SelectTableColumnExtendLeft 709
- SelectTableColumnExtendRight 709
- SelectTableRow 710
- SelectTabularColumn 710
- SelectToggle 710
- SelectWord 710
- SelectWordNext 711
- SelectWordPrevious 711
- SENDKEYS 216
- SENDKEYS WAIT 216
- SetDefaultFileFormat 711
- SetDocumentMapping, one-byte characters 713
- SettingsDlg 713
- SetupWritingToolsDlg 714
- Shadow, in border *see* BorderDropShadow
- Shadowed text *see* Font
- Shared Code 4, 970
- ShiftTabKey 714
- ShowBoxes 714
- ?ShowBoxes 954
- ShowHiddenText 715
- ShowSymbols 715
- Shut *see* Close
- Silhouette of font character, turn on/off 435
- SoftHyphen 715
- SoftSpace 716
- Sort 716
  - case priority, specifying 717
  - character string, insert into Select Records option 718
  - key options, specifying 718
  - source file format, specifying 719
  - whole file or certain records only 716
- SortAction 716
- SortCaseOrder 717
- SortDlg 717
- ?SortGetAction 955
- ?SortGetCase 955
- SortKeys 718
- SortSelectString 718
- SortType 719
- Sound board, causing to beep 80
- Sound clip
  - attributes, setting 721
  - creating 719
  - deleting 720
  - description, setting 722
  - filename, setting 722
  - specific clip to play, specifying 721
- Sound file, playing 189
- SoundClipCreate 719
- SoundClipDelete 720
- SoundClipDlg 721
- SoundClipPlayNext 721
- SoundClipSetAttributes 721
- SoundClipSetDescription 722
- SoundClipSetFilename 722
- Source files 5
- ?SpaceExpansion 955
- Spacing
  - automatic, selecting 270
  - between words and letters 861
  - inside a border, specifying 265
  - kerning 502
  - outside a border, specifying 267
  - paragraph, space between, specifying 616
- Spacing between characters *see* Kerning
- ?SpacingBelowPar 955
- Speaker, causing computer to beep 80
- SpecificDeleteStringBegin 723
- SpecificReplaceStringBegin 723
- SpecificSearchStringBegin 724
- SpecificStringEnd 724
- Speech file
  - playing 190
  - speaking from clipboard 191
  - stopping 192
- SPEED 219
- Spell checker, opening 863
- Spreadsheet files, specifying location of 650
- SpreadsheetShowLinkCode 725
- SpreadsheetUpdateOnRetrieve 725
- SQL *see* NetWare SQL
- ?StartingLineNumber 956
- Statements
  - calling 28
  - conditional 25
  - Loop 26
- Static control, adding to a dialog box 140
- Status Bar preferences
  - default, setting 661
  - editing 662
  - options, specifying 662
- ?StatusBarActive 956
- StatusBarShow 726
- String functions
  - STRLEN 220
  - STRNUM 222
- STRPOS 223
- STRUNIT 224
- SUBSTR 227
- UNITSTR 232
- Strings, entering in an edit box 172
- STRLEN 220
- STRNUM 222
- STRPOS 223
- STRUNIT 224
- Style *see also* Box style; Line style; Outline style
  - box border, specifying 279
  - box fill, specifying 303
  - box, copying 319
  - button bar, specifying 335
  - character, turning off 738
  - codes, editing 727
  - copying 727
  - creating 730
  - deleting 730
  - description of, changing 732
  - edit, beginning 733
  - edit, ending 735
  - fill, creating 415
  - graphics box, changing 287
  - graphics line, modifying 461
  - key setting, entering 736
  - renaming 742
  - retrieving 742
  - saving 743
  - system or user-defined, turning on 744
  - system, turning on 740
  - turning on with Enter 736
  - type, specifying 746
- Style, border *see also* Border style
  - bottom line, specifying 260
  - copying 270
  - deleting 272
  - editing 273
  - editing, end 274
  - left line, specifying 266
  - naming new 272
  - renaming 275
  - retrieving 275
  - right line, specifying 268
  - saving 276
  - separator in paragraph/columns, specifying 269
  - specifying 830
  - top line, specifying 276
- StyleCodes 727
- StyleCopy 727

StyleCreate 730  
StyleDelete 730  
StyleDescription 732  
StyleEditBegin 733  
StyleEditEnd 735  
StyleEnterKeySetting 736  
StyleLinkTo 736  
StyleOff 738  
StyleOn 740  
StyleRename 742  
StyleRetrieve 742  
StyleSave 743  
StylesListDlg 744  
StyleSystemOn 744  
StyleType 746  
SUBCHAR 226  
SubdividePage 746  
SubdividePageDlg 747  
SubdividePageOff 747  
SubdocInclude 747  
Subdocument  
    condensing 545  
    currently active 907  
    expanding 546  
Subroutine name 28, 29  
Subroutines 9  
    calling 89  
    creating 168, 182, 208  
    FUNCTION 168  
    LABEL 182  
    PROCEDURE 208  
    returning from 213  
    sharing functions and procedures 234  
Subscript 250  
SUBSTR 227  
?Substructure 956  
SubstructureExit 748  
Summary *see* Document summary  
Superscript 250  
Suppress 748  
SWITCH 228  
SWITCH-ENDSWITCH 26, 228  
SwitchDoc 749  
Syntax 6  
System Variables 7

## T

Tab 749  
    Hard Back Tab, inserting 252  
    Hard Left Tab, inserting 800  
    Hard Right Tab, inserting 822  
    settings, specifying 823  
        underlining, specifying 852  
TabAdd 749  
TabCenter 750  
TabDecimal 751  
TabDelete 751  
TabInsert 752  
TabKey *see also*  
Table *see also* Insertion point, movement  
    in tables  
        adjust position, beginning 752  
        adjust position, ending 753  
        adjust position, specifying 754  
        Automatic Calculation Mode,  
            specifying 756  
        character attributes, turning off 754  
        character attributes, turning on 755  
        column break, inserting 801  
        currency symbols in, aligning 809  
        currency type in, specifying 813  
        current table name 908  
        date format in, specifying 810  
        deleting 796  
        fill style, specifying 797  
        formula, placing in cell 792  
        guides, turning on/off 801  
        header row, designating 802  
        joining tables 804  
        justification in, specifying 798  
        last item moved/copied, retrieving  
            816  
        name of, deleting 809  
        name, assigning new 808  
        page break, inserting 802  
        position of, specifying 815  
        Select, turning off 757, 819  
        Select, turning on 757, 819  
        splitting into two 820  
        text in tabular/parallel columns,  
            converting to 789  
        undeleting 822  
Table cell *see also* Cell; Cell attribute;  
    Floating cell  
    block, turning off 819  
    block, turning on 819  
    bottom margin, specifying 818  
    character attributes, turning off 760  
    character attributes, turning on 761  
    column attribute, using in 773  
    column justification, using in 774  
    column number type, using in 774  
    commas, displaying 772  
    contents, converting to formula 821

    contents, converting to tab/merge  
        codes 796  
    contents, copying 762, 789  
    contents, deleting 794  
    contents, moving 806  
    currency symbol, displaying with  
        numbers in 773  
    currency symbols, aligning in 767  
    currency type, specifying 771  
    date format, specifying 768  
    decimals in, converting to integers  
        770  
    default line color, specifying 775  
    digits after decimal in, specifying 769  
    foreground/background colors of,  
        specifying 762  
    ignoring when calculating 764  
    joining to another 805  
    justification, specifying 765  
    left margin, specifying 805  
    line color around, specifying 775  
    line color, specifying 766  
    line options, specifying 765  
    lock, turning off 797  
    locking to prevent editing 767  
    name, specifying 807  
    negative number display, specifying  
        770  
    number format for, specifying 769  
    placing formula/data in 792  
    right margin, specifying 806  
    shading style, specifying 763  
    splitting into multiple columns 820  
    splitting into multiple rows 821  
    sum, calculating 816  
    top margin, specifying 818  
    vertical alignment, specifying 775  
Table column  
    block, turning off 819  
    block, turning on 819  
    break, inserting 801  
    character attribute, turning off 776  
    character attribute, turning on 777  
    contents, copying 789  
    contents, moving 806  
    currency symbols, aligning 781  
    date format, specifying 781  
    deleting one or more 795  
    fixed width, designating 779  
    incrementing values in 756  
    inserting 803  
    justification, specifying 779



- left edge and text, distance between 780
- name, specifying 807
- right edge and text, distance between 780
- size, decreasing 788
- size, increasing 788
- width, specifying 753, 787, 799
- width, specifying as fixed 787
- Table Formula Control Bar, display 800
- Table Guides, turning on/off 801
- Table of Authorities
  - control bar, displaying 843
  - definition, creating 836
  - definition, deleting 838
  - definition, inserting 833
  - format, specifying 664
  - full form, editing 843
  - generating 448
  - item, marking 544
  - marking blocked text as reference 845
  - page format, changing 844
  - page references display, changing 839
  - reference to specific citation, creating 545
  - renaming section 839
  - retrieving 840
  - section, specifying which to edit 844
  - sequential page numbers, combining 835
  - short form, marking 846
  - short form, renaming 845
  - style, changing 840
  - underlining, allowing 842
- Table of Contents
  - blocked text, marking 848
  - defining/inserting definition 846
  - generating 448
  - marking 545
- Table row
  - block, turning off 819
  - block, turning on 819
  - contents, copying 789
  - contents, moving 806
  - deleting one or more 796
  - height, specifying as fixed/automatic 817
  - incrementing values in 756
  - inserting new above current 803
  - inserting new below current 754
  - inserting specific number of 804
  - move insertion point down one 619
  - name, specifying 807
- Table, formulas in *see also* Formulas
  - converting from cell contents 821
  - copying to current cell 790
  - entering at document level/edit mode 757
  - inserting in current cell 799
  - Table Formula Control Bar,
    - displaying 800
    - values, calculating 759
    - values, calculating for all in document 760
- Table, lines in
  - border edit, beginning 758
  - border edit, ending 759
  - border type, specifying 758
  - color of, specifying as default 822
  - color, specifying as default 794
  - grids, turning on/off 801
  - multiple, in cells 819
  - style, setting default 793
- Table, numbers in *see also* Formulas
  - commas, displaying 772, 786, 814
  - currency symbols in columns,
    - displaying 786
    - currency symbols, displaying 815
    - currency type, specifying 785
    - decimals to integers in columns,
      - converting 784
    - decimals to integers in table cells,
      - converting 770
    - decimals to integers in tables,
      - converting 812
    - digits after decimal point in columns,
      - specifying 782
    - digits after decimal point, specifying 810
    - digits between decimal and right edge of cell, spe 792
    - distance from right edge of cell,
      - specifying 792
    - format for current column, specifying 783
    - incrementing values in row/column 756
    - negative number display in columns,
      - specifying 783
    - negative number display, specifying 812
    - number format, specifying 811
    - sum of cells, calculating 816
  - TableAdjustBegin 752
  - TableAdjustColumnWidth 753
  - TableAdjustEnd 753
  - TableAdjustPosition 754
  - TableAppendRow 754
  - TableAttributeOff 754
  - TableAttributeOn 755
  - TableAutoCalc 756
  - TableAutoFill 756
  - TableAutoFormulaEntry 757
  - TableBlockOff 757
  - TableBlockOn 757
  - TableBorder 758
  - TableBorderEditBegin 758
  - TableBorderEditEnd 759
  - TableCalculate 759
  - TableCalculateAll 760
  - TableCalculateDlg 760
  - TableCellAttributeOff 760
  - TableCellAttributeOn 761
  - TableCellCopy 762
  - TableCellFillColors 762
  - TableCellFillStyle 763
  - TableCellIgnoreCalculation 764
  - TableCellJustification 765
  - TableCellLine 765
  - TableCellLineColor 766
  - TableCellLock 767
  - TableCellNumberAlignCurrency 767
  - TableCellNumberDateFormat 768
  - TableCellNumberDecimalDigits 769
  - TableCellNumberFormat 769
  - TableCellNumberNegativeNumber 770
  - TableCellNumberRound 770
  - TableCellNumberSelectCurrency 771
  - TableCellNumberUseCommas 772
  - TableCellNumberUseCurrency 773
  - TableCellUseColumnAttribute 773
  - TableCellUseColumnJust 774
  - TableCellUseColumnNumberType 774
  - TableCellUseLineColor 775
  - TableCellVerticalAlignment 775
  - TableColumnAttributeOff 776
  - TableColumnAttributeOn 777
  - TableColumnDecimalAlignDigits 778
  - TableColumnDecimalAlignDist 778
  - TableColumnFixedWidth 779, 787
  - TableColumnJustification 779
  - TableColumnMarginLeft 780
  - TableColumnMarginRight 780
  - TableColumnNumberAlignCurrency 781
  - TableColumnNumberDateFormat 781
  - TableColumnNumberDecimalDigits 782
  - TableColumnNumberFormat 783

TableColumnNumberNegNumber 783  
 TableColumnNumberRound 784  
 TableColumnNumberSelCurrency 785  
 TableColumnNumberUseCommas 786  
 TableColumnNumberUseCurrency 786  
 TableColumnWidth 787  
 TableColumnWidthGrow 788  
 TableColumnWidthShrink 788  
 TableConvertDlg 788  
 TableConvertFrom 789  
 TableCopy 789  
 TableCopyFormula 790  
 TableCreate 791  
 TableCreateDlg 791  
 TableCreateNewDlg 791  
 TableDataFill 792  
 TableDecimalAlignmentDigits 792  
 TableDecimalAlignmentDistance 792  
 TableDefaultLine 793  
 ?TableDefaultLine 956  
 TableDefaultLineColor 794  
 TableDeleteBlock 794  
 TableDeleteColumn 795  
 TableDeleteDlg 795  
 TableDeleteRow 796  
 TableDeleteTable 796  
 TableDisableCellLocks 797  
 TableEdit 797  
 TableFill 797  
 TableFormatDlg 798  
 TableFormatJustification 798  
 TableFormatWidths 799  
 TableFormula 799  
 TableFormulaCBar 800  
 TableLeft 800  
 TableGridLinesToggle 801  
 TableGuides 801  
 ?TableGuides 957  
 TableHardColumnBreak 801  
 TableHardRow 802  
 TableHeader 802  
 TableInsertColumn 803  
 TableInsertDlg 803  
 TableInsertOneRow 803  
 TableInsertRow 804  
 ?TableInTable 957  
 TableJoin 804  
 TableJoinCells 805  
 ?TableLastCell 957  
 TableLineFillDlg 805  
 TableMarginLeft 805  
 TableMarginRight 806  
 TableMove 806  
 TableMoveCopyDlg 807  
 TableNameAdd 807  
 TableNameChange 808  
 TableNameCreateDlg 808  
 TableNameDelete 809  
 TableNameListDlg 809  
 TableNumberAlignCurrency 809  
 TableNumberDateFormat 810  
 TableNumberDecimalDigits 810  
 TableNumberFormat 811  
 TableNumberNegativeNumber 812  
 TableNumberRound 812  
 TableNumberSelectCurrency 813  
 TableNumberTypeDlg 814  
 TableNumberUseCommas 814  
 TableNumberUseCurrency 815  
 TablePosition 815  
 TableQuickSum 816  
 TableRetrieve 816  
 TableRowHeight 817  
 TableRowMarginBottom 818  
 TableRowMarginTop 818  
 TableRowNumberOfLines 819  
 TableSelectOff 819  
 TableSelectOn 819  
 TableSplit 820  
 TableSplitCellsDlg 820  
 TableSplitColumn 820  
 TableSplitRow 821  
 TableTextToFormula 821  
 TableUndelete 822  
 TableUseDefaultLinesColor 822  
 TabRight 822  
 TabSet 823  
 TabSetDlg 823  
 ?TabSettings 957  
 Targets, naming *see* the CrossRef  
   commands  
 Task list, display 246  
 Template  
   abbreviation in, creating 241  
   abbreviation in, deleting 241  
   files, specifying location of 650  
 TemplateCopyObjectDlg 824  
 TemplateCreate 824  
 TemplateCreateDlg 824  
 TemplateDescriptionDlg 825  
 TemplateDlg 825  
 TemplateEdit 825  
 ?TemplateFile 958  
 ?TemplateFileSupplemental 958  
 TemplateSelect 826  
 Text  
   adding to Clipboard 244  
   bookmark, creating 257  
   bookmark, deleting 258  
   bookmark, finding 259  
   bookmark, finding/blocking as 257  
   bookmark, moving 259  
   bookmark, renaming 260  
   indenting without changing margins  
     493  
   inserting at insertion point 849  
   keeping together on page 859  
   position on page, specifying 243  
   typecover, turning on/off 850  
 Text appearance  
   attributes, turning on/off 248  
   bold, turning on/off 257  
   centering one line 337  
   color, specifying 831  
   color, specifying intensity 832  
   font size, specifying 250  
   Hidden Text option, turning on/off  
     473  
   hiding/showing hidden text 473  
   justification options, specifying 499  
   modifying 453  
   normal, restoring to 249  
   position on page, specifying 243  
   positioning as sub/superscript 249  
   reducing/enlarging size 665  
 Text at bottom of page *see* Footer  
 Text at top of page *see* Header  
 Text border  
   creating 826  
   editing 827  
   end editing/saving changes 828  
 Text box, creating 831  
 Text control, adding to a dialog box 140  
 Text in boxes *see* BoxCaption and  
   BoxText commands  
 Text/graphics in boxes *see* Box  
   commands  
 TextBorderCreate 826  
 TextBorderEdit 827  
 TextBorderEnd 828  
 TextBorderFillStyle 829  
 TextBorderStyle 830  
 TextBoxCreate 831  
 TextColor 831  
 ?TextColorBlue 958  
 ?TextColorGreen 958  
 ?TextColorRed 958  
 ?TextColumnNumber 959  
 ?TextColumns 959

- TextShade 832
- ?TextShade 959
- Thesaurus, display options 863
- ThousandsSeparator 832
- ?ThousandsSeparatorChar 959
- Timed Backup, minutes between 253
- Timed Document Backup, turn on/off 252
- ?TimeHour 959
- ?TimeHundredth 960
- ?TimeMinute 960
- ?TimeSecond 960
- ToA *see* Table of Authorities
- ToACreateFullFormDlg 833
- ToADefineDlg 833
- ToADefinition 833
- ?ToADefinition 960
- ToADefinitionCombine 835
- ToADefinitionCreate 836
- ToADefinitionDelete 838
- ToADefinitionMode 839
- ToADefinitionRename 839
- ToADefinitionRetrieve 840
- ToADefinitionStyle 840
- ToADefinitionUnderline 842
- ToAEditFullForm 843
- ToAEditFullFormCBar 843
- ToAEditFullFormDlg 843
- ToAEditPageFormat 844
- !oAEditSection 844
- ToAEditShortForm 845
- ToAMark 845
- ToAMarkShortForm 846
- ?ToAShortFormMarks 960
- ?ToAShortForms 961
- ToC *see* Table of Contents
- ToCDefine 846
- ToCDefineDlg 848
- ?ToCDefinition 961
- ToCMark 848
- TOLOWER 230
- TOUPPER 231
- Type 849
- TypeChar 849
- Typeover 850
- ?TypeoverActive 961
- TypeoverKey 850
- TypesetManualKerningDlg 851

**U**

- Unary Operators 12
- Undelete 851
  - last deletion 853
  - up to last three deletions 851
- ?UndeleteBuffer 961
- UndeleteDlg 851
- Underline, turning on/off 438
- UnderlineSpaces 852
- ?UnderlineSpaces 961
- UnderlineTabs 852
- ?UnderlineTabs 962
- Undo 853
- Unit of Measure
  - entering in an edit box 173
  - setting default 106
- UNITSTR 232
- ?UnprintableBottomMargin 962
- ?UnprintableLeftMargin 962
- ?UnprintableRightMargin 962
- ?UnprintableTopMargin 963
- UNTIL 212
- ?UpdateQuickList 963
- Uppercase *see* Convert case
- USE 31, 234
- ?UseDefaultDocExt 963
- User-defined items, display menu 184
- ?UserColorBlue 963
- ?UserColorGreen 963
- ?UserColorRed 964
- UserFunction 853
- ?UserInitials 964
- ?UserName 964

**V**

- VARERRCHK 236
- Variable, assigning data to 450
- Variables
  - assigning values to 79
  - creating global 174
  - creating local 104, 183
  - creating names with INDIRECT 179
  - creating persist 206, 207
  - details 10
  - determining if exists 161
  - determining type 161
  - explanation of 7
  - removing from memory 154
  - responding to uninitialized 236
- Variables, Constants, Operators, and Expressions 10

- Vertical line control, adding to a dialog box 143
- View document
  - as full page 609, 864
  - between width of margins 864
  - between width of page 864
  - in draft mode, at 100% 389
  - in draft mode, to margins 390
  - text/graphics, reducing or enlarging size of 665
  - two pages at once 854
  - with all features 854
  - without features (as draft) 853
- View preferences
  - features display, setting 639
  - Outline control bar, displaying 854
  - Power Bar, displaying 636
  - Reveal Codes, displaying 687
  - Ruler Bar, displaying 687
- View window *see also* Window
  - left side of, displaying 688
  - maximize size of 244
  - minimize size of 244
  - repositioning 245
  - resizing 246
  - restoring to previous size 245
  - right side of, displaying 688
- ViewDraft 853
- Viewer control, adding to a dialog box 142
- ViewOutline 854
- ViewPage 854
- ViewTwoPage 854
- VLineCreate 855
- Volume number
  - decreasing by one 855
  - displaying 856
  - increasing by one 856
  - specifying 855
  - specifying numbering type 856
- VolumeNumber 855
- ?VolumeNumber 964
- VolumeNumberDecrement 855
- VolumeNumberDisplay 856
- VolumeNumberIncrement 856
- VolumeNumberMethod 856
- ?VolumeNumberMethod 964

**W**

- WAIT 236 *see also* SENDKEYS WAIT
- Watermark
  - creating/editing/turning off A 857

creating/editing/turning off B 857  
 WatermarkA 857  
 ?WatermarkAOccur 965  
 WatermarkB 857  
 ?WatermarkBOccur 965  
 WatermarkDlg 858  
 WatermarkPlacementDlg 858  
 WHILE 237  
 WHILE-ENDWHILE 27, 237  
 WidowOrphan 859  
 ?WidowOrphan 965  
 Window *see also* View window  
   activating 73  
   displaying side by side 860  
   enlarging 244  
   left side, viewing 688  
   opening next 859  
   opening previous 860  
   overlapping windows so title bars  
     display 859  
   reducing 244  
   repositioning 245  
   resizing 246  
   restoring to previous size 245  
   right side, viewing 688  
 WindowCascade 859  
 WindowNext 859  
 WindowPrevious 860  
 Windows Task List, displaying 246  
 WindowTile 860  
 Word spacing, adjusting for full  
   justification 861  
 WordCountDlg 860  
 WordLetterspaceDlg 860  
 WordLetterSpacing 861  
 WordPerfect character, insert 849  
 WordPerfect, specifying if it's running  
   on network 942  
 WordSpacingJustificationLimits 861  
 WPCharactersDlg 862  
 WPDRAW 862  
 WPGrammatik 862  
 WPSpeller 863  
 WPTThesaurus 863  
 Writing tools, turning on/off 863  
 WritingToolsDisable 863  
 ?WritingToolsDisabledState 966

enlarging/reducing size of  
   text/graphics 665  
   full view of current page, displaying  
     609  
 ?Zoom 966  
 ZoomDlg 864  
 ZoomToFullPage 864  
 ZoomToMarginWidth 864  
 ZoomToPageWidth 864

## Z

ZIP Code, specifying 254  
 Zoom

---

# The WordPerfect 6.0 for Windows Software Developer's Kit

Your assignment: develop a set of solid add-in products that have the look and feel of a WordPerfect product—and that integrate with WordPerfect software. Sound tedious? It doesn't have to be—if you have the right tools!

---

## The Tools You Need for the Job

Everything you need to speed development time, increase functionality, and integrate with WordPerfect software is now available in one convenient package! The WordPerfect 6.0 for Windows Software Developer's Kit comes complete with sample code, documentation, and all the information you need to use WordPerfect's innovative tools.

### **Develop Add-in Features with Ease**

New tools in this kit give you total flexibility in developing add-in features to WordPerfect applications. You can directly manipulate the text in WPCorp applications—*without* having to understand WPCorp formatting codes!

### **Customize WordPerfect with Your Own Code**

Any functionality you add will look like part of the original WPCorp application. The sample code in the kit shows you how to add menu items, create custom commands, and display command IDs. We even make it easy for your DLLs to block, change, or ignore WordPerfect commands—all without sacrificing speed or risking accidental termination!

### **Use the Power of Shared Code**

With the software developer's kit, you'll save development time and give your applications the look and feel of a WPCorp product with Shared Code 2.0 for Windows, a shared library of routines used by all WPCorp products. Sample code in the kit teaches you how to register with Shared Code, making full integration not only possible, but easy!

The WordPerfect 6.0 for Windows Software Developer's Kit is as easy to order as it is to use. Simply phone or fax:

Telephone Orders (800) 321-4566

Fax Orders (801) 228-5377

If you are outside the United States, Puerto Rico, the U.S. Virgin Islands, or Canada, please contact your local WordPerfect affiliate or authorized reseller.