PARADOX FOR WINDOWS

QUICK REFERENCE

BORLAND

Paradox for Windows

Version 1.0

Quick Reference

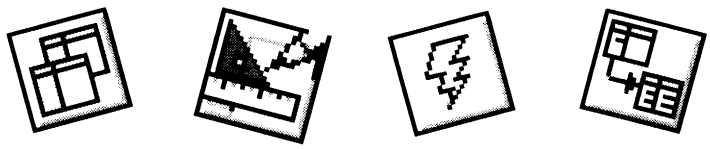
Borland International 1800 Green Hills Road
P.O. Box 660001, Scotts Valley, CA 95067-0001, USA

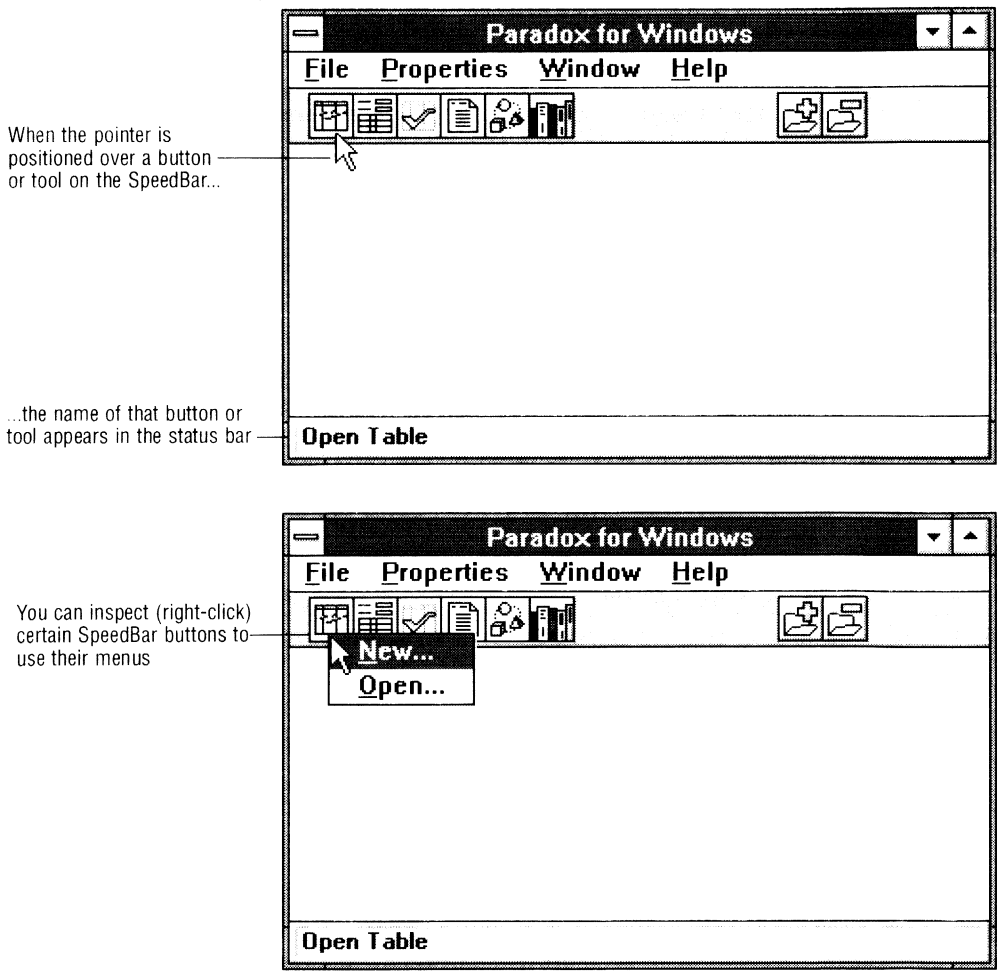
10 9 8 7 6 5 4 3 2 1 R1

CONTENTS

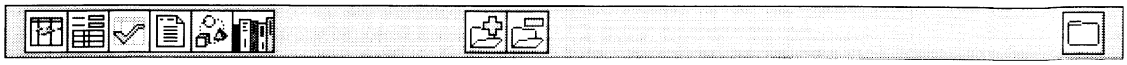
PART I

SpeedBar buttons and tools





Desktop SpeedBar



Open Table: Open a table. Right-click to create a table.

Open Form: Open a form. Right-click to create a form.

Open Query: Open a query. Right-click to create a query.

Open Report: Open or print a report. Right-click to create a report.

Open Script: Open or run an ObjectPAL script. Right-click to create a script.

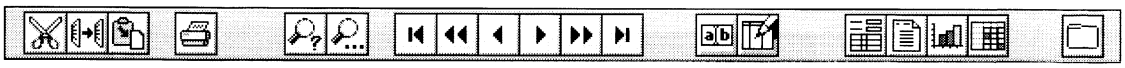
Open Library: Open an ObjectPAL library. Right-click to create a library.

Add Folder Item: Add an object's icon to the Folder window. The Folder window doesn't need to be open.

Remove Folder Item: Remove an object's icon from the Folder window. Removing an icon does not delete the object.

Open Folder: Open the working directory's Folder window.

Table window's SpeedBar



Cut to Clipboard: Delete the selected data and place it on the Clipboard.

Copy to Clipboard: Place a copy of the selected data on the Clipboard.

Paste from Clipboard: In Edit mode, insert the current Clipboard contents into the selected field.

Print: Print a report of the table's data.

Locate Field Value: Locate a specific value in the field.

Locate Next: Locate the next occurrence of the value you entered with Locate Field Value.

First Record: Move to the first record.

Previous Record Set: Scroll up one screen.

Previous Record: Move up one record.

Next Record: Move down one record.

Next Record Set: Scroll down one screen.

Last Record: Move to the last record.

Field View: Enter or exit Field View to move character-by-character through a field.

Edit Data: Enter or exit Edit mode to add or modify data.

Quick Form: View the table in a Form window.

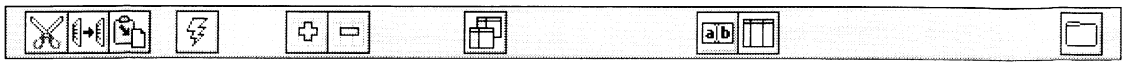
Quick Report: Preview a report of the table's data in the Report window.

Quick Graph: View the table's data in a graph.

Quick Crosstab: View the table's data in a crosstab.

Open Folder: Open the working directory's Folder window.

Query window's SpeedBar



Cut to Clipboard: Delete the selected text and place it on the Clipboard. You can cut example elements or text you type in query statements.

Copy to Clipboard: Place a copy of the selected text on the Clipboard. You can copy example elements or text you type in query statements.

Paste from Clipboard: Insert the current Clipboard contents into the selected field.

Run Query: Execute the query and display the results.

Add Table: Add a table to the Query window.

Remove Table: Remove a table from the Query window.

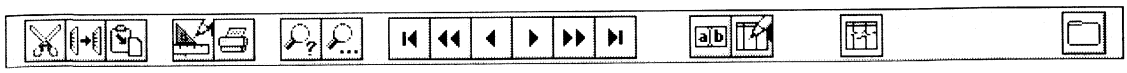
Join Tables: Join two tables in a multi-table query. Use the mouse to place example elements.

Field View: Enter or exit Field View to move character-by-character through a field.

Answer Table Properties: Change the name, table type, or properties of the *Answer* table before you run the query.

Open Folder: Open the working directory's Folder window.

Form window's SpeedBar



Cut to Clipboard: Delete the selected data and place it on the Clipboard.

Copy to Clipboard: Place a copy of the selected data on the Clipboard.

Paste from Clipboard: In Edit mode, insert the current Clipboard contents into the selected field.

Design: Open the Form Design window, where you can change the form's layout and properties.

Print: Print the form one record at a time.

Locate Field Value: Locate a specific value in the field.

Locate Next: Locate the next occurrence of the value you entered with Locate Field Value.

First Record: Move to the first record.

Previous Record Set: Move up 10 records. In a multi-record form, move up one screenful of records.

Previous Record: Move up one record.

Next Record: Move down one record.

Next Record Set: Move down 10 records. In a multi-record form, move down one screenful of records.

Last Record: Move to the last record.

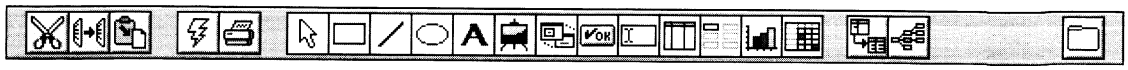
Field View: Enter or exit Field View to move character-by-character through a field.

Edit Data: Enter or exit Edit mode to add or modify data.

Table View: Open the form's master table in a Table window.

Open Folder: Open the working directory's Folder window.

Form Design window's SpeedBar



Cut to Clipboard: Delete the selected data and place it on the Clipboard.

Copy to Clipboard: Place a copy of the selected data on the Clipboard.

Paste from Clipboard: In Edit mode, insert the current Clipboard contents into the selected field.

View Data: Run the form. This lets you view or edit data.

Print: Print the design (without data) of the form.

Selection Arrow: Select objects to move, resize, cut, or copy them.

Box tool: Create a box. Press and hold *Shift* to create multiple boxes.

Line tool: Create a line. Press and hold *Shift* to create multiple lines.

Ellipse tool: Create an ellipse. Press and hold *Shift* to create multiple ellipses.

Text tool: Create a text object. Press and hold *Shift* to create multiple text objects. To create a fixed-size text object, drag a frame. To create a variable-size text object, click in the design area and begin typing.

Graphic tool: Create a graphic object. Press and hold *Shift* to create multiple graphic objects. To paste into the frame, choose Edit | Paste or Edit | Paste From.

OLE tool: Create an OLE object. Press and hold *Shift* to create multiple OLE objects. To paste data into the object, first copy it from the OLE server.

Button tool: Create a button object. Press and hold *Shift* to create multiple buttons. Attach an ObjectPAL method to the button so users can click it to perform a task.

Field tool: Create a field object. Press and hold *Shift* to create multiple field objects.

Table tool: Create a table frame to display a table. Press and hold *Shift* to create multiple table frames.

Multi-Record tool: Create a multi-record object to display several records of a table. Press and hold *Shift* to create multiple multi-record objects.

Graph tool: Create a graph object to display a graph. Press and hold *Shift* to create multiple graph objects.

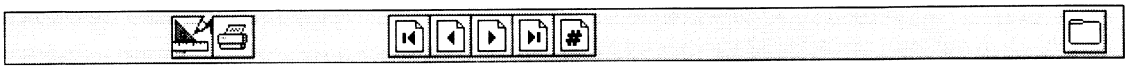
Crosstab tool: Create a crosstab object. Press and hold *Shift* to create multiple crosstab objects.

Data Model: View the form's data model. You can specify the tables to bind to the form and how the tables are linked to each other.

Object Tree: View the diagram of the hierarchy of objects in the form.

Open Folder: Open the working directory's Folder window.

Report window's SpeedBar



Design: Open the Report Design window, where you can change the report's layout and properties.

Print: Print the report. You can print the current page or the entire report.

First Page: Move to the first page.

Previous Page: Move up one page.

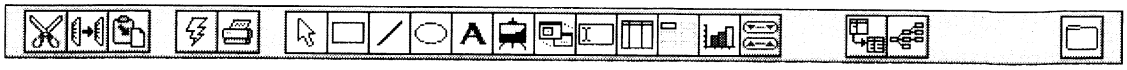
Next Page: Move down one page.

Last Page: Move to the last page.

Go To Page: Move to a specified page.

Open Folder: Open the working directory's Folder window.

Report Design window's SpeedBar



Cut to Clipboard: Delete the selected data and place it on the Clipboard.

Copy to Clipboard: Place a copy of the selected data on the Clipboard.

Paste from Clipboard: In Edit mode, insert the current Clipboard contents into the selected field.

View Data: Run the report to preview it onscreen.

Print: Print the design (without data) of the report.

Selection Arrow: Select objects to move, resize, cut, or copy them.

Box tool: Create a box. Press and hold *Shift* to create multiple boxes.

Line tool: Create a line. Press and hold *Shift* to create multiple lines.

Ellipse tool: Create an ellipse. Press and hold *Shift* to create multiple ellipses.

Text tool: Create a text object. Press and hold *Shift* to create multiple text objects. To create a fixed-size text object, drag a frame. To create a variable-size text object, click in the design area and begin typing.

Graphic tool: Create a graphic object. Press and hold *Shift* to create multiple graphic objects. To paste into the frame, choose Edit | Paste or Edit | Paste From.

OLE tool: Create an OLE object. Press and hold *Shift* to create multiple OLE objects. To paste data into the object, first copy it from the OLE server.

Field tool: Create a field object. Press and hold *Shift* to create multiple field objects.

Table tool: Create a table frame to display a table. Press and hold *Shift* to create multiple table frames.

Multi-Record tool: Create a multi-record object to display several records of a table. Press and hold *Shift* to create multiple multi-record objects.

Graph tool: Create a graph object to display a graph. Press and hold *Shift* to create multiple graph objects.

Add Band: Add a group band to the report to organize data into groups.

Data Model: View the report's data model. You can specify the tables to bind to the report and how the tables are linked to each other.

Object Tree: View the diagram of the hierarchy of objects in the report.

Open Folder: Open the working directory's Folder window.

PART II

Reference tables

Table 1 Menu shortcuts

Key	Action in a Table window	Action in any design window
Alt+Backspace	Undo	
Ctrl+A	Locate Next	Locate Next
Ctrl+D	Ditto (repeat field from previous record)	Ditto (repeat field from previous record)
Ctrl+F	Field View	Field View
Ctrl+G	Inspect Grid	
Ctrl+H	Inspect Heading	
Ctrl+Ins	Copy	Copy
Ctrl+Shift+H	Inspect All Headings	Inspect All Headings
Ctrl+L	Lock Record	Lock Record
Ctrl+Shift+L	Post Record	Post Record
Ctrl+M	Inspect Field	Object Inspector
Ctrl+Shift+M	Inspect All Fields	
Ctrl+R	Rotate Columns	Rotate Columns (on table object)
Ctrl+T	Memo View	Memo View
Ctrl+Z	Locate Value	Locate Value
Ctrl+Shift+Z	Locate and Replace	Locate and Replace
Del	Clear or Delete (as appropriate)	Clear or Delete (as appropriate)
Shift+Del	Cut	Cut
Shift+Ins	Paste	Paste

Table 2 Function keys

Key	Action in a table	Action in a form	Action in a query
F1	Help	Help	Help
F2	Field View	Field View	Field View
Shift+F2	Memo View (&DDE/OLE)	Memo View (&OLE)	
Ctrl+F2	Persistent Field View	Persistent Field View	Persistent Field View
F3		Super Back Tab	Up Image
Shift+F3		Page Back	
F4		Super Tab	Down Image
Shift+F4		Page Forward	
F5	Lock Record	Lock Record	Example
Shift+F5	Post Record	Post Record	
Ctrl+F5	Post/Keep Lock	Post/Keep Lock	
F6	Object Inspector	Object Inspector	Checkmark (✓)
Shift+F6	Inspect All	Penetrating Properties	Cycle checks (✓, ✓+, ✓↓, ✓**G**)
F7	Quick Form	Table View	
Shift+F7	Quick Report		
Ctrl+F7	Quick Graph		
F8		View/Design	Run Query
F9	Edit/End Edit	Edit/End Edit*	
F10	Menu	Menu	Menu
F11	Previous Record	Previous Record	
Shift+F11	Previous Set	Previous Set	
Ctrl+F11	First Record	First Record	
F12	Next Record	Next Record	
Shift+F12	Next Set	Next Set	
Ctrl+F12	Last Record	Last Record	

* If you press *F9* in the Form Design window, Paradox opens the form in Edit mode. This is a shortcut to pressing *F8* (View Data) followed by *F9* (Edit Data).

Table 3 Navigation and selection keys

Key	Non-Field View	Field View
PgUp	Up one set of records	Up one set of records
Ctrl+PgUp	Left one screen	Left one screen
PgDn	Down one set of records	Down one set of records
Ctrl+PgDn	Right one screen	Right one screen
Home	First field of record	Beginning of field
Shift+Home	Select to first field of record*	Select to beginning of field
Ctrl+Home	First field of first record	First field of first record
Alt+Home	First field of record	First field of record
End	Last field of record	End of field
Shift+End	Select to last field of record*	Select to end of field
Ctrl+End	Last field of last record	Last field of last record
Alt+End	Last field of record	Last field of record
←	Left one field	Left one character
Shift ←	Select left one field*	Select left one character
Ctrl ←	First column	Left one word
Ctrl+Shift ←	Select to first column of table	Extend selection left one word
Alt ←	Left one field	Left one field
→	Right one field	Right one character
Shift →	Select right one field*	Select right one character
Ctrl →	Last column	Right one word
Ctrl+Shift →	Select to last column of table	Extend selection right one word
Alt →	Right one field	Right one field
↑	Up one field	Up one line in multi-line field or up one record in single-line field
Shift ↑	Select up one field*	Select up one line within multi-line field or up one record in single-line field
Alt ↑	Up one field	Up one field
↓	Down one field	Down one line within multi-line field or down one record in single-line field
Shift ↓	Select down one field*	Select down one line within multi-line field or down one record in single-line field
Alt ↓	Down one field	Down one field

* You can select multiple fields only in tables, not in forms.
Note Be sure *Num Lock* is *off* when you use *Alt* in combination with a keypad key.

Table 4 Keys to use in Edit mode

Key	Action
Ins	Insert record
Shift+Ins	Paste
Ctrl+Ins	Copy
Del	Delete selected text
Shift+Del	Cut
Ctrl+Del	Delete record
Backspace	Delete character to the left or delete selected text
Ctrl+Backspace	Delete word to left
Alt+Backspace	Undo record edit
Esc	Undo field edit
Tab	Post value and move to next field
Shift+Tab	Post value and move to previous field
Enter	Post value and move to next field

Table 5 Query operators

Category	Operator	Meaning
Reserved symbols	✓	Display unique field values in *Answer*
	✓+	Display field values including duplicates in *Answer*
	✓↓	Display unique field values in descending order
	✓**G**	Specify a group for set operations
Reserved words	CALC	Calculate a new field and display results in *Answer*
	INSERT	Insert records with specified values and create a temporary table
	DELETE	Remove records with specified values and create a temporary table
	CHANGETO	Change specified values in fields and create a temporary table
	SET	Define specific records as a set for comparisons
Arithmetic operators	+	Addition or alphanumeric string concatenation
	–	Subtraction
	*	Multiplication
	/	Division
	( )	Group operators in a query expression

Category	Operator	Meaning
Comparison operators	=	Equal to (optional)
	>	Greater than
	<	Less than
	>=	Greater than or equal to
	<=	Less than or equal to
Wildcard operators	..	Any series of characters
	@	Any single character
Special operators	LIKE	Similar to
	NOT	Does not match
	BLANK	No value
	TODAY	Today's date
	OR	Specify OR conditions in a field
	,	Specify AND conditions in a field
	AS	Specify the name of a field in *Answer*
	!	Include record in *Answer* even if not matched in joined table
Summary operators	AVERAGE	Average of values in a field
	COUNT	Number of values in a field
	MAX	Highest value in a field
	MIN	Lowest value in a field
	SUM	Total of all values in a field
	ALL	Calculate summary based on all values in a group, including duplicates
	UNIQUE	Calculate summary based on unique values in a group
Set comparison operators	ONLY	Display records that match only members of the defined set
	NO	Display records that match no members of the defined set
	EVERY	Display records that match every member of the defined set
	EXACTLY	Display records that match all members of the defined set and no others

Table 6 File extensions

Extension	Type of object
.CFG files	Configuration file
.DB files	Paradox table
.DBF files	dBASE table
.DBT files	Memos for a dBASE table
.FAM files	Paradox's listing of related files (like a table's .TV file)
.FDL files	Delivered form
.FSL files	Saved form
.FTL files	Temporary form
.INI files	Configuration file
.LDL files	Delivered library
.LSL files	Saved library
.LTL files	Temporary library
.MB files	Memos for a Paradox table
.MDX files	Maintained index of a dBASE table
.NDX files	Non-maintained index of a dBASE table
.PX files	Primary index of a Paradox table
.QBE files	Saved query
.RDL files	Delivered report
.RSL files	Saved report
.RTL files	Temporary report
.SDL files	Delivered script
.SSL files	Saved script
.STL files	Temporary script
.TV files	Table view settings for a Paradox table
.TVF files	Table view settings for a dBASE table
.VAL files	Validity checks and referential integrity for a Paradox table
.X*nn* files	Secondary single-field index for a Paradox table, numbered
.Y*nn* files	Secondary single-field index for a Paradox table, numbered
.XG*n* files	Composite secondary index for a Paradox table
.YG*n* files	Composite secondary index for a Paradox table

Table 7 Temporary table names

Name	Contains	Created during
Answer	Results from a query	Query
Changed	Unchanged copy of changed records	CHANGETO query or Add operation (update)
Crosstab	Results of a crosstab	Running a crosstab object in a form
Deleted	Deleted records	DELETE query
Errchng	Records that couldn't be changed	CHANGETO query
Errdel	Records that couldn't be deleted	DELETE query
Errins	Records that couldn't be inserted	INSERT query
Export	The table specification of exported fixed-length text	Export operation
Import	The table specification of imported fixed-length text	Import operation
Inserted	Inserted records	INSERT query
*Keyviol**	Records with duplicate key values	Restructure *or* Add operation (append)
Locks	All active locks on a table	File|Multiuser|Display Locks
Pal$src	List of source code, objects, and methods in your form	Language|Browse Sources
*Problems**	Unconverted records	File|Utilities|Import *or* Restructure
Struct	Table definition	Create *or* Restructure

* If you perform more than one operation that results in this temporary table within one session, Paradox creates additional temporary tables with the same name and numbers them. For example, *Keyviol1, Keyviol2,...*

PART III

ObjectPAL syntax

abs

Number *Method*
Returns the absolute value of a number.
abs ( ) Number

accessRights

FileSystem *Method*
Reports access rights (also called file attributes) of a file.
accessRights ( ) String

acos

Number *Method*
Returns the 2-quadrant arc cosine of a number.
acos ( ) Number

action

Form *Method/Procedure*
Performs an action command.
action (const ***actionId*** SmallInt) Logical

TableView *Method*
Performs an action command.
action (const ***actionId*** SmallInt) Logical

UIObject *Method*
Performs an action command.
action (const ***actionId*** SmallInt) Logical

actionClass

ActionEvent *Method*
Returns the class number of an ActionEvent.
actionClass ( ) SmallInt

add

Table *Method*

Adds the data in one table to another table.

1. add (const ***destTableName*** String
[, const ***append*** Logical [, const ***update*** Logical]]) Logical

2. add (const ***destTable*** Table
[, const ***append*** Logical [, const ***update*** Logical]]) Logical

Compatibility Procedure

1. add (const ***tableName*** String ,
const ***destTableName*** String) Logical

2. add (const ***tableName*** String ,
const ***destTableName*** String [, const ***append*** Logical
[, const ***update*** Logical]]) Logical

TCursor *Method*

Adds the records of one table to another.

1. add (const ***destTableName*** String [, const ***append*** Logical
[, const ***update*** Logical]]) Logical

2. add (const ***destTCursor*** TCursor [, const ***append*** Logical
[, const ***update*** Logical]]) Logical

3. add (const ***destTable*** Table [, const ***append*** Logical
[, const ***update*** Logical]]) Logical

addAlias

Session *Method/Procedure*

Adds a database alias to a session.

addAlias (const ***aliasName*** String, const ***type*** String,
const ***path*** String) Logical

addArray

Menu *Method*

Appends elements of an array to a menu.

addArray (const ***items*** Array[] String)

PopUpMenu *Method*

Appends elements of an array to a pop-up menu.

addArray (const ***items*** Array[] String)

addBar

PopUpMenu *Method*

Adds a vertical bar to a pop-up menu.

addBar ( )

addBreak

Menu *Method*
Starts a new row in a menu.
addBreak ( )

PopUpMenu *Method*
Starts a new column in a pop-up menu.
addBreak ( )

addLast

Array *Method*
Inserts an item at the end of a resizeable array.
addLast (const ***value*** AnyType)

addPassword

Session *Method/Procedure*
Presents a password allowing access to a protected table.
addPassword (const ***password*** String)

addPopUp

Menu *Method*
Adds a pop-up menu to a menu bar item.
addPopUp (const ***menuName*** String, const ***cascadedPopUp*** PopUpMenu)

PopUpMenu *Method*
Adds a pop-up menu to the structure.
addPopUp (const ***menuName*** String, const ***cascadedPopUp*** PopUpMenu)

addSeparator

PopUpMenu *Method*
Adds a horizontal bar to a pop-up menu.
addSeparator ( )

addStaticText

Menu *Method*
Adds an unselectable text string to a menu.
addStaticText (const ***item*** String)

PopUpMenu *Method*
Adds an unselectable text string to a pop-up menu.
addStaticText (const ***item*** String)

addText

Menu *Method*

Adds a selectable text string to a menu.

1. **addText** (const ***menuName*** String)
2. **addText** (const ***menuName*** String, const ***attrib*** SmallInt)
3. **addText** (const ***menuName*** String, const ***attrib*** SmallInt, const ***id*** SmallInt)

PopUpMenu *Method*

Adds a selectable text string to a pop-up menu.

1. **addText** (const ***menuName*** String)
2. **addText** (const ***menuName*** String, const ***attrib*** SmallInt)
3. **addText** (const ***menuName*** String, const ***attrib*** SmallInt, const ***id*** SmallInt)

advancedWildcardsInLocate

Session *Procedure*

Specifies whether this session can use advanced wildcards in locate operations.

advancedWildcardsInLocate ([const ***yesNo*** Logical])

advMatch

String *Method*

Searches text for a specified string.

advMatch (const ***pattern*** String [, var ***matchVar*** String]*) Logical

TextStream *Method*

Searches for a pattern of characters in a text file.

advMatch (var ***startIndex*** LongInt, var ***endIndex*** LongInt, const ***pattern*** String) Logical

ansiCode

String *Procedure*

Returns the ANSI code of a one-character string.

ansiCode (const ***char*** String) SmallInt

append

Array *Method*

Appends the contents of one array to another.

append (const ***newArray*** Array[] AnyType)

asin

Number — *Method*
Returns the 2-quadrant arc sine of a number.
asin ( ) Number

atan

Number — *Method*
Returns the 2-quadrant arc tangent of a number.
atan ( ) Number

atan2

Number — *Method*
Returns the 4-quadrant arc tangent of a number.
atan2 (const ***x*** Number) Number

atFirst

TCursor — *Method*
Reports whether the TCursor is pointing to the first record of a table.
atFirst ( ) Logical

UIObject — *Method*
Reports whether the pointer is at the first record of a table.
atFirst ( ) Logical

atLast

TCursor — *Method*
Reports whether the TCursor is pointing to the last record of a table.
atLast ( ) Logical

UIObject — *Method*
Reports whether the pointer is at the last record in a table.
atLast ( ) Logical

attach

Form — *Method*
Associates a Form variable with an open form.
attach ([const ***formName*** String]) Logical

Report — *Method*
Associates a Report variable with an open report.
attach (const ***reportTitle*** String) Logical

Table *Method*
Associates a Table variable with a table on disk.

1. **attach** (const ***tableName*** String) Logical
2. **attach** (const ***tableName*** String, const ***db*** Database) Logical
3. **attach** (const ***tableName*** String, const ***tableType*** String) Logical
4. **attach** (const ***tableName*** String, const ***tableType*** String, const ***db*** Database) Logical

TCursor *Method*
Binds a TCursor to a UIObject.

1. **attach** (const ***object*** UIObject) Logical
2. **attach** (const ***srcTCursor*** TCursor) Logical
3. **attach** (const ***tv*** TableView) Logical

UIObject *Method*
Binds a UIObject variable to a specified design object.

1. **attach** ( ) Logical
2. **attach** (const ***object*** UIObject) Logical
3. **attach** (const ***form*** Form [, const ***objectName*** String]) Logical
4. **attach** (const ***form*** Report [, const ***objectName*** String]) Logical

attachToKeyViol

TCursor *Method*
Attaches a TCursor to the existing record that has the same key as the record you attempted to post.
attachToKeyViol (const ***oldTC*** TCursor) Logical

beep

System *Procedure*
Sounds the Windows default beep.
beep ( )

bitAND

LongInt *Method*
Performs a bitwise AND operation on two values.
bitAND (const ***value*** LongInt) LongInt

SmallInt *Method*
Performs a bitwise AND operation on two values.
bitAND (const ***value*** SmallInt) SmallInt

bitIsSet

LongInt *Method*
Reports whether a bit is 1 or 0.
bitIsSet (const ***value*** LongInt) Logical

SmallInt *Method*
Reports whether a bit is 1 or 0.
bitIsSet (const ***value*** SmallInt) Logical

bitOR

LongInt *Method*
Performs a bitwise OR operation on two values.
bitOR (const ***value*** LongInt) LongInt

SmallInt *Method*
Performs a bitwise OR operation on two values.
bitOR (const ***value*** SmallInt) SmallInt

bitXOR

LongInt *Method*
Performs a bitwise XOR operation on two values.
bitXOR (const ***value*** LongInt) LongInt

SmallInt *Method*
Performs a bitwise XOR operation on two values.
bitXOR (const ***value*** SmallInt) SmallInt

blank

AnyType *Method*
Returns a blank value.
blank ( )

Procedure
blank ( ) AnyType

blankAsZero

Session *Method/Procedure*
Specifies whether to treat blank values as zeros in calculations.
blankAsZero (const ***yesNo*** Logical)

bot

TCursor *Method*
Tests for a move past the beginning of a table.
bot ( ) Logical

breakApart

String *Method*
Splits a string into substrings.
breakApart (var ***tokenArray*** Array[] String [,const ***separators*** String])

bringToTop

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*
Brings a window to the top of the display stack and makes it active.
bringToTop ( )

broadcastAction

UIObject *Method*
Broadcasts an action to an object.
broadcastAction (const ***actionID*** SmallInt)

cancelEdit

TCursor *Method*
Ends Edit mode without saving changes to the current record.
cancelEdit ( ) Logical

UIObject *Method*
Cancels record changes without ending Edit mode.
cancelEdit ( ) Logical

canReadFromClipboard

OLE *Method*
Reports whether an OLE object can be pasted from the Clipboard into an OLE variable.
canReadFromClipboard ( ) Logical

cAverage

Table *Method*
Returns the average value of a field (column) in a table.
1. **cAverage** (const ***fieldName*** String) Number
2. **cAverage** (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. **cAverage** (const ***tableName*** String, const ***fieldName*** String) Number
2. **cAverage** (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the average value of a field (column) in a table.
1. cAverage (const ***fieldName*** String) Number

2. cAverage (const ***fieldNum*** SmallInt) Number

cCount

Table *Method*
Returns the number of nonblank values in a field (column) of a table.
1. cCount (const ***fieldName*** String) Number

2. cCount (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cCount (const ***tableName*** String, const ***fieldName*** String) Number

2. cCount (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the number of values in a field (column) of a table.
1. cCount (const ***fieldName*** String) Number

2. cCount (const ***fieldNum*** SmallInt) Number

ceil

Number *Method*
Rounds a numeric expression up to the nearest whole number.
ceil ( ) Number

char

KeyEvent *Method*
Returns the character associated with a keypress.
char ( ) String

charAnsiCode

KeyEvent *Method*
Returns the ANSI value associated with a keypress.
charAnsiCode ( ) SmallInt

chr

String *Procedure*
Returns the one-character string represented by an ANSI code.
chr (const ***ansiCode*** SmallInt) String

chrOEM

String *Procedure*
Returns the one-character string of an OEM code.
chrOEM (const ***oemCode*** SmallInt) String

chrToKeyName

String *Procedure*
Returns the virtual key-code string of a one-character string.
chrToKeyName (const ***char*** String) String

close

Database *Method*
Closes a database.
close ( ) Logical

DDE *Method*
Closes a DDE link.
close ( ) Logical

Form *Method*
Closes a Form window.
close ( )

Procedure
close ([const ***returnValue*** AnyType])

Library *Method*
Closes a library.
close ( )

Report *Method*
Closes a Report window.
close ( )

Session *Method*
Closes a session.
close ( ) Logical

System *Procedure*
Closes the current form.
close ([const ***returnValue*** AnyType])

TableView *Method*
Closes a Table window.
close ( )

TCursor *Method*
Closes a TCursor.
close ( ) Logical

TextStream *Method*
Closes a text file.
close ( ) Logical

cMax

Table *Method*
Returns the maximum value of a field (column) in a table.
1. cMax (const ***fieldName*** String) Number

2. cMax (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cMax (const ***tableName*** String, const ***fieldName*** String) Number

2. cMax (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the maximum value of a field (column) in a table.
1. cMax (const ***fieldName*** String) Number

2. cMax (const ***fieldNum*** SmallInt) Number

cMin

Table *Method*
Returns the minimum value in a field (column) of a table.
1. cMin (const ***fieldName*** String) Number

2. cMin (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cMin (const ***tableName*** String, const ***fieldName*** String) Number

2. cMin (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the minimum value in a field (column) of a table.
1. cMin (const ***fieldName*** String) Number

2. cMin (const ***fieldNum*** SmallInt) Number

cNpv

Table *Method*

Returns the net present value of a field (column), based on a specifed discount or interest rate.

1. **cNpv** (const ***fieldName*** String, const ***discRate*** AnyType) Number
2. **cNpv** (const ***fieldNum*** SmallInt, const ***discRate*** AnyType) Number

Compatibility Procedure

1. **cNpv** (const ***tableName*** String, const ***fieldName*** String, const ***discRate*** AnyType) Number
2. **cNpv** (const ***tableName*** String, const ***fieldNum*** SmallInt, const ***discRate*** AnyType) Number

TCursor *Method*

Returns the net present value of a field (column), based on a specified discount or interest rate.

1. **cNpv** (const ***fieldName*** String, const ***discRate*** Number) Number
2. **cNpv** (const ***fieldNum*** SmallInt, const ***discRate*** Number) Number

commit

TextStream *Method*

Writes the contents of the text buffer to disk.

commit ( )

compact

Table *Method*

Removes deleted records from a table.

compact ([const ***regIndex*** Logical]) Logical

TCursor *Method*

Removes deleted records from a table.

compact ([const ***regIndex*** Logical]) Logical

constantNameToValue

System *Procedure*

Returns the numeric value of a constant.

constantNameToValue (const ***constantName*** String) AnyType

constantValueToName

System *Procedure*

Reports on the name of a constant.

constantValueToName (const ***groupName*** String, const ***value*** AnyType, var **constName** String) Logical

contains

Array *Method*
Searches the items of an Array for a pattern of characters.
contains (const ***value*** AnyType) Logical

DynArray *Method*
Searches the indexes in a DynArray for a value.
contains (const ***value*** AnyType) Logical

Menu *Method*

PopUpMenu *Method*
Reports whether an item is in a menu or pop-up menu.
contains (const ***item*** AnyType) Logical

convertPointWithRespectTo

UIObject *Method*
Changes the frame of reference for calculating the coordinates of a point.
convertPointWithRespectTo (const ***otherUIObject*** UIObject, const ***oldPoint*** Point, var ***convertedPoint*** Point)

copy

FileSystem *Method*
Copies a file.
copy (const ***srcName*** String, const ***destName*** String) Logical

Table *Method*
Copies a table.
1. copy (const ***destTableName*** String) Logical

2. copy (const ***destTableVar*** Table) Logical

Compatibility Procedure
1. copy (const ***tableName*** String, const ***destTableName*** String) Logical

2. copy (const ***tableName*** String, const ***destTableVar*** Table) Logical

TCursor *Method*
Copies a table.
1. copy (const ***destTableName*** String) Logical

2. copy (const ***destTableVar*** Table) Logical

copyFromArray

TCursor *Method*
Copies data from an array to a record of a table.
1. copyFromArray (const ***ar*** Array[] AnyType) Logical

2. copyFromArray (const ***ar*** DynArray[] AnyType) Logical

UIObject *Method*
Copies data from an array to a record of a table.
copyFromArray (const ***ar*** Array[] AnyType) Logical

copyRecord

TCursor *Method*
Copies a record pointed to by one TCursor into the record pointed to by another TCursor.
copyRecord (const ***pointer*** TCursor) Logical

copyToArray

TCursor *Method*
Copies the fields of the current record to an array.
1. copyToArray (var ***ar*** Array[] AnyType) Logical

2. copyToArray (var ***ar*** DynArray[] AnyType) Logical

UIObject *Method*
Copies data from a record to an array.
copyToArray (var ***ar*** Array[] AnyType) Logical

cos

Number *Method*
Returns the cosine of an angle.
cos ( ) Number

cosh

Number *Method*
Returns the hyperbolic cosine of an angle.
cosh ( ) Number

count

Menu *Method*

PopUpMenu *Method*
Returns the number of items in a menu or pop-up menu.
count ( ) SmallInt

countOf

Array *Method*

Counts the occurrences of a value in an array.

countOf (const ***value*** AnyType) LongInt

cpuClockTime

System *Procedure*

Returns the number of seconds since the computer was started.

cpuClockTime ( ) LongInt

create

Form *Method*

Report *Method*

Creates a blank form or report in a design window.

create ( ) Logical

Table *Keyword*

Creates a table.

create "*tableName*" [**as** "*tableType*"] [**Database** *db*]
  [[**like** *likeObject*]
    [**with** "*fieldName*" : "*type*" [, "*fieldName*" : "*type*"] *]
    [**where** *fieldDesc* **is** "*newName*" [, *fieldDesc* **is** "*newname*"] *]
    [**without** *fieldDesc* [, *fieldDesc*] *]
    [**struct** *fieldStructTable*]
    [**indexStruct** *indexStructTable*]
    [**refIntStruct** *refIntStructTable*]
    [**secStruct** *secStructTable*]
    ] *
  [**key** *fieldDesc* [, *fieldDesc*] *]
endCreate

TextStream *Method*

Creates a text file for reading and writing.

create (const ***fileName*** String) Logical

UIObject *Method*

Creates an object.

create (const ***objectType*** SmallInt, const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt, const ***h*** LongInt [, const ***container*** UIObject])

cSamStd

Table *Method*
Returns the sample standard deviation of a field (column) of a table.
1. cSamStd (const ***fieldName*** String) Number

2. cSamStd (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cSamStd (const ***tableName*** String, const ***fieldName*** String) Number

2. cSamStd (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the sample standard deviation of a field (column) of a table.
1. cSamStd (const ***fieldName*** String) Number

2. cSamStd (const ***fieldNum*** SmallInt) Number

cSamVar

Table *Method*
Returns the sample variance of a field (column) in a table.
1. cSamVar (const ***fieldName*** String) Number

2. cSamVar (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cSamVar (const ***tableName*** String, const ***fieldName*** String) Number

2. cSamVar (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the sample variance of a field (column) in a table.
1. cSamVar (const ***fieldName*** String) Number

2. cSamVar (const ***fieldNum*** SmallInt) Number

cStd

Table *Method*
Returns the population standard deviation of a field (column) in a table.
1. cStd (const ***fieldName*** String) Number

2. cStd (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cStd (const ***tableName*** String, const ***fieldName*** String) Number

2. cStd (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the population standard deviation of a field (column) in a table.
1. cStd (const ***fieldName*** String) Number

2. cStd (const ***fieldNum*** SmallInt) Number

cSum

Table *Method*
Returns the sum of the values in a field (column) of a table.
1. cSum (const ***fieldName*** String) Number

2. cSum (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cSum (const ***tableName*** String, const ***fieldName*** String) Number

2. cSum (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the sum of the values in a field (column) of a table.
1. cSum (const ***fieldName*** String) Number

2. cSum (const ***fieldNum*** SmallInt) Number

currency

Currency *Method*
Casts a value as Currency.
currency (const ***value*** AnyType) Currency

currentPage

Report *Method*
Returns the current page number of a report.
currentPage ( ) SmallInt

currRecord

TCursor *Method*
Reads the current record into the record buffer.
currRecord ( ) Logical

UIObject *Method*
Reads the current record into the record buffer.
currRecord ( ) Logical

cVar

Table *Method*
Returns the variance of a field in a table.
1. cVar (const ***fieldName*** String) Number

2. cVar (const ***fieldNum*** SmallInt) Number

Compatibility Procedure
1. cVar (const ***tableName*** String, const ***fieldName*** String) Number

2. cVar (const ***tableName*** String, const ***fieldNum*** SmallInt) Number

TCursor *Method*
Returns the variance of a field (column) in a table.
1. cVar (const ***fieldName*** String) Number

2. cVar (const ***fieldNum*** SmallInt) Number

data

MenuEvent *Method*
Returns information about a MenuEvent.
data ( ) LongInt

dataType

AnyType *Method*
Returns a string representing the data type of a variable.
dataType ( ) String

date

Date *Procedure*
Casts a value as a Date or returns the current date.
date ([const ***value*** AnyType]) Date

dateTime

DateTime *Method*
Casts a value as a DateTime or returns the current date and time.
dateTime ([const ***value*** AnyType]) DateTime

dateVal

Date *Procedure*
Returns a value as a date.
dateVal (const ***value*** AnyType) Date

day

Date *Method*

DateTime *Method*
Extracts the day of the month from a Date or DateTime.
day ( ) SmallInt

daysInMonth

Date *Method*

DateTime *Method*
Returns the number of days in a month.
daysInMonth ( ) SmallInt

debug

System *Procedure*
Halts execution of a method and invokes the Debugger.
debug ( )

delayScreenUpdates

Form *Procedure*
Turns delayed screen updates on or off.
delayScreenUpdates (const ***yesNo*** Logical)

delete

Database *Method/Procedure*
Deletes a table from a database.
1. delete (const ***tableName*** String [, const ***tableType*** String]) Logical

2. delete (const ***tableVar*** Table) Logical

FileSystem *Method*
Deletes a file.
delete (const ***name*** String) Logical

Table *Method*
Deletes a table.
delete ( ) Logical

Compatibility Procedure
delete (const ***tableName*** String [, const ***tableType*** String]) Logical

UIObject *Method*
Deletes an object from a form.
delete ( )

deleteDir

FileSystem *Method*
Deletes a directory.
deleteDir (const ***name*** String) Logical

deleteRecord

TCursor *Method*
Deletes the record pointed to by a TCursor.
deleteRecord ( ) Logical

UIObject *Method*
Deletes the current record from the table.
deleteRecord ( ) Logical

deliver

Form *Method*
Delivers a form.
deliver ( ) Logical

design

Form *Method*
Switches a running form to the Form Design window.
design ( ) Logical

Report *Method*
Switches a running report to the Report Design window.
design ( ) Logical

didFlyAway

TCursor *Method*
Reports whether the current record moved to a different position as the result of a key value change.
didFlyAway ( ) Logical

disableBreakMessage

Form *Procedure*
Prevents program interruption by *Ctrl+Break*.
disableBreakMessage (const ***yesNo*** Logical) Logical

distance

Point *Method*
Returns the distance between two points.
distance (const ***pt*** Point) Number

dlgAdd

System *Procedure*
Invokes the Table Add dialog box.
dlgAdd (const ***tableName*** String)

dlgCopy

System *Procedure*
Invokes the Table Copy dialog box.
dlgCopy (const ***tableName*** String)

dlgCreate

System *Procedure*
Invokes the Create Table dialog box.
dlgCreate (const ***tableName*** String)

dlgDelete

System *Procedure*
Invokes the Table Delete dialog box.
dlgDelete (const ***tableName*** String)

dlgEmpty

System *Procedure*
Invokes the Table Empty dialog box.
dlgEmpty (const ***tableName*** String)

dlgNetDrivers

System *Procedure*
Invokes the Current Drivers dialog box.
dlgNetDrivers ( )

dlgNetLocks

System *Procedure*
Creates and displays a table of lock information.
dlgNetLocks ( )

dlgNetRefresh

System *Procedure*
Invokes the Network Refresh Rate dialog box.
dlgNetRefresh ( )

dlgNetRetry

System *Procedure*
Invokes the Network Retry Period dialog box.
dlgNetRetry ( )

dlgNetSetLocks

System *Procedure*
Invokes the Table Locks dialog box.
dlgNetSetLocks ( )

dlgNetSystem

System *Procedure*
Invokes the ODAPI System Information dialog box.
dlgNetSystem ( )

dlgNetUserName

System *Procedure*
Invokes the Network User Name dialog box.
dlgNetUserName ( )

dlgNetWho

System *Procedure*
Invokes the Current Users dialog box.
dlgNetWho ( )

dlgRename

System *Procedure*
Invokes the Table Rename dialog box.
dlgRename (const ***tableName*** String)

dlgRestructure

System *Procedure*
Invokes the Table Restructure dialog box.
dlgRestructure (const ***tableName*** String)

dlgSort

System *Procedure*
Invokes the Table Sort dialog box.
dlgSort (const ***tableName*** String)

dlgSubtract

System *Procedure*
Invokes the Table Subtract dialog box.
dlgSubtract (const ***tableName*** String)

dlgTableInfo

System *Procedure*
Invokes the Structure Information dialog box.
dlgTableInfo (const ***tableName*** String)

dmAddTable

Form *Method/Procedure*
Adds a table to a form's data model.
dmAddTable (const ***tableName*** String) Logical

dmGet

Form *Method/Procedure*
Retrieves a field value from a table in the data model.
dmGet (const ***tableName*** String, const ***fieldName*** String, var ***datum*** AnyType) Logical

dmHasTable

Form *Method/Procedure*
Reports whether a table is part of a form's data model.
dmHasTable (const ***tableName*** String) Logical

dmPut

Form *Method/Procedure*
Writes data to a table in a form's data model.
dmPut (const ***tableName*** String, const ***fieldName*** String, const ***datum*** AnyType) Logical

dmRemoveTable

Form *Method/Procedure*
Removes a table from a form's data model.
dmRemoveTable (const ***tableName*** String) Logical

dow

Date *Method*
DateTime *Method*
Returns the day of the week of a Date or DateTime.
dow ( ) String

dowOrd

Date *Method*
DateTime *Method*
Returns the number of a day of the week.
dowOrd ( ) SmallInt

doy

Date *Method*
DateTime *Method*
Returns the number of a day of the year.
doy ( ) SmallInt

drives

FileSystem *Method*
Returns the letters of the drives attached to the system and known to Windows.
drives ( ) String

dropIndex

Table *Method*
Deletes an index file associated with a table.
1. (Paradox tables) **dropIndex** ([const ***indexName*** String]) Logical
2. (dBASE tables) **dropIndex** ([const ***indexName*** String [, const ***tagName*** String]]) Logical

TCursor *Method*
Deletes an index file associated with a table.
1. (Paradox tables) **dropIndex** ([const ***indexName*** String]) Logical
2. (dBASE tables) **dropIndex** (const ***indexName*** String [, const ***tagName*** String]) Logical

edit

OLE *Method*
Launches the OLE server and lets the user edit the object or take some other action.
edit (const ***oleText*** String, const ***verb*** SmallInt) Logical

TCursor *Method*
Puts a TCursor into Edit mode.
edit ( ) Logical

UIObject *Method*
Puts a table into Edit mode.
edit ( ) Logical

empty

Array *Method*
Removes all items from an array.
empty ( )

DynArray *Method*
Removes all items from a dynamic array.
empty ( )

Menu *Method*

PopUpMenu *Method*
Removes all items from a menu or pop-up menu.
empty ( )

Table *Method*
Deletes all records from a table.
empty ( ) Logical

Compatibility Procedure
empty (const ***tableName*** String) Logical

TCursor *Method*
Deletes all records from a table.
empty ( ) Logical

UIObject *Method*
Deletes all records from a table.
empty ( ) Logical

end

TCursor *Method*
Moves a TCursor to the last record in a table.
end ( ) Logical

TextStream *Method*
Moves the file pointer to the end of a text file.
end ( )

UIObject *Method*
Moves the pointer to the last record in a table.
end ( ) Logical

endEdit

TCursor *Method*
Exits Edit mode and accepts changes to the current record.
endEdit ( ) Logical

UIObject *Method*
Exits Edit mode and accepts changes to the current record.
endEdit ( ) Logical

enumAliasNames

Session *Method/Procedure*
Creates a Paradox table listing the names of database aliases available to a session.
enumAliasNames (const ***tableName*** String) Logical

enumDatabaseTables

Session *Method/Procedure*
Creates a Paradox table listing the tables in a database.
enumDataBaseTables (const ***tableName*** String,
const ***databaseName*** String, const ***fileSpec*** String)

enumDesktopWindowNames

System *Procedure*

Creates a table listing open Paradox windows.

1. **enumDesktopWindowNames** (const ***tableName*** String)
2. **enumDesktopWindowNames**
   (const ***windowNames*** Array[] String)

enumDriverCapabilities

Session *Procedure*

Creates three Paradox tables that list the capabilities of the current driver.

enumDriverCapabilities (const ***drvCapName*** String,
const ***tblCapName*** String, const ***fldCapName*** String)

enumDriverInfo

Session *Procedure*

Lists the available drivers.

enumDriverInfo (const ***tableName*** String)

enumDriverNames

Session *Procedure*

Creates a Paradox table listing the names of the drivers available in the current session.

enumDriverNames (const ***tableName*** String)

enumDriverTopics

Session *Procedure*

Creates a Paradox table listing the topics currently available for each driver type.

enumDriverTopics (const ***tableName*** String)

enumEngineInfo

Session *Procedure*

Creates Paradox table listing the current ODAPI engine properties.

enumEngineInfo (const ***tableName*** String)

enumFieldNames

Table *Method*

Fills an array with the names of fields in a table.

enumFieldNames (var ***fieldArray*** Array[] String) Logical

TCursor *Method*

Fills an array with the names of fields in a table.

enumFieldNames (var ***fieldArray*** Array[] String) Logical

UIObject *Method*
Fills an array with the names of fields in a table.
enumFieldNames (var ***fieldNames*** Array[] String) Logical

enumFieldNamesInIndex

Table *Method*
Fills an array with the names of fields in a table's index.
1. (Paradox tables) **enumFieldNamesInIndex**
([const ***indexName*** String,] var ***fieldArray*** Array[] String) Logical
2. (dBASE tables) **enumFieldNamesInIndex**
([const ***indexName*** String, [const ***tagName*** String,]]
var ***fieldArray*** Array[] String) Logical

TCursor *Method*
Fills an array with the names of fields in a table's index.
1. (Paradox tables) **enumFieldNamesInIndex**
([const ***indexName*** String,] var ***fieldArray*** Array[] String) Logical
2. (dBASE tables) **enumFieldNamesInIndex**
([const ***indexName*** String [, const ***tagName*** String ,]
var ***fieldArray*** Array[] String) Logical

enumFieldStruct

Table *Method*
Creates a Paradox table listing the structure of a table.
enumFieldStruct (const ***tableName*** String) Logical

TCursor *Method*
Creates a Paradox table listing the structure of a TCursor.
enumFieldStruct (const ***tableName*** String) Logical

enumFileList

FileSystem *Method*
Writes information about files to a table or an array.
1. **enumFileList** (const ***fileSpec*** String, var ***arrayName*** Array[] String)
2. **enumFileList** (const ***fileSpec*** String, const ***tableName*** String)

enumFolder

Session *Procedure*
Creates a Paradox table or array listing files in a folder.
1. **enumFolder** (const ***tableName*** String [, const **fileSpec** String]) Logical
2. **enumFolder** (var ***result*** Array[] String [, const **fileSpec** String]) Logical

enumFonts

System *Procedure*
Creates a table listing fonts in the user's system.
enumFonts (const ***tableName*** String)

enumFormNames

System *Procedure*
Creates an array listing open forms.
enumFormNames (var ***formNames*** Array[] String)

enumIndexStruct

Table *Method*
Creates a Paradox table listing the structure of a table's secondary indexes.
enumIndexStruct (const ***tableName*** String) Logical

TCursor *Method*
Creates a Paradox table listing the structure of a TCursor's secondary indexes.
enumIndexStruct (const ***tableName*** String) Logical

enumLocks

TCursor *Method*
Creates a Paradox table listing the locks currently applied to a TCursor; returns number of locks.
enumLocks (const ***tableName*** String) LongInt

UIObject *Method*
Creates a Paradox table listing the locks currently applied to a UIObject; returns number of locks.
enumLocks (const ***tableName*** String) LongInt

enumObjectNames

UIObject *Method/Procedure*
Fills an array with the names of the objects in a form.
enumObjectNames (var ***objectNames*** Array[] String)

enumOpenDatabases

Session *Method/Procedure*
Creates a Paradox table listing the open databases.
enumOpenDatabases (const ***tableName*** String) Logical

enumRefIntStruct

Table *Method*
Creates a Paradox table listing a table's referential integrity information.
enumRefIntStruct (const ***tableName*** String) Logical

TCursor *Method*
Creates a Paradox table listing a TCursor's referential integrity information.
enumRefIntStruct (const ***tableName*** String) Logical

enumReportNames

System *Procedure*
Creates an array listing open reports.
enumReportNames (var ***reportNames*** Array[] String)

enumRTLClassNames

System *Procedure*
Creates a table listing the object types in ObjectPAL.
enumRTLClassNames (const ***tableName*** String) Logical

enumRTLConstants

System *Procedure*
Creates a table listing the constants defined by ObjectPAL.
enumRTLConstants (const ***tableName*** String) Logical

enumRTLMethods

System *Procedure*
Creates a table listing the methods in ObjectPAL.
enumRTLMethods (const ***tableName*** String) Logical

enumSecStruct

Table *Method*
Creates a Paradox table listing a table's security information.
enumSecStruct (const ***tableName*** String) Logical

TCursor *Method*
Creates a Paradox table listing a TCursor's security information.
enumSecStruct (const ***tableName*** String) Logical

enumSource

Form *Method/Procedure*
Creates a table listing the methods for each object in a form.
enumSource (const ***tableName*** String [, const ***recurse*** Logical]) Logical

Library *Method*
Writes the code from a library to a Paradox table.
enumSource (const ***tableName*** String [, const ***recurse*** Logical]) Logical

UIObject *Method*
Fills a table with the source code of the methods on a form.
enumSource (const ***tableName*** String [, const ***recurse*** Logical]) Logical

enumSourceToFile

Form *Method/Procedure*
Creates a file listing the code for a form.
enumSourceToFile (const ***fileName*** String
[, const ***recurse*** Logical]) Logical

Library *Method*
Creates a file listing the code for a library.
enumSourceToFile (const ***fileName*** String
[, const ***recurse*** Logical]) Logical

UIObject *Method*
Creates a file listing the code for an object.
enumSourceToFile (const ***fileName*** String
[, const ***recurse*** Logical]) Logical

enumTableLinks

Form *Method/Procedure*
Creates a table listing the tables linked to a form.
enumTableLinks (const ***tableName*** String) Logical

enumTableProperties

TCursor *Method*
Writes the properties of a TCursor to a Paradox table.
enumTableProperties (const ***tableName*** String) Logical

enumUIClasses

UIObject *Procedure*
Writes a list of UIObject classes to a table.
enumUIClasses (const ***tableName*** String) Logical

enumUIObjectNames

Form *Method*
Creates a table listing the UIObjects contained in a form.
enumUIObjectNames (const ***tableName*** String) Logical

Report *Method*
Creates a table listing the UIObjects contained in a report.
enumUIObjectNames (const ***tableName*** String) Logical

UIObject *Method/Procedure*
Gets the names of each object in a form and writes them to a table.
enumUIObjectNames (const ***tableName*** String) Logical

enumUIObjectProperties

Form *Method*
Creates a table listing the properties of each UIObject contained in a form.
enumUIObjectProperties (const ***tableName*** String) Logical

Report *Method*
Creates a table listing the properties of each UIObject contained in a report.
enumUIObjectProperties (const ***tableName*** String) Logical

UIObject *Method/Procedure*
Creates a table listing the properties of each UIObject contained in a form or report.
enumUIObjectProperties (const ***tableName*** String) Logical

enumUsers

Session *Procedure*
Creates a Paradox table listing all known users with an open channel to the ODAPI engine.
enumUsers (const ***tablename*** String) LongInt

enumVerbs

OLE *Method*
Creates a DynArray listing the actions supported by an OLE server.
enumVerbs (var ***verbs*** DynArray[] SmallInt) Logical

enumWindowNames

System *Procedure*
Creates a table or an array listing open windows for the system.
1. **enumWindowNames** (const ***tableName*** String) Logical
2. **enumWindowNames** (var ***windowNames*** Array[] String)

eof

TextStream *Method*
Tests for a move past the end of a text file.
eof ( ) Logical

eot

TCursor *Method*
Tests for a move past the end of a table.
eot ( ) Logical

errorClear

System *Procedure*
Clears the error stack.
errorClear ( )

errorCode

ActionEvent *Method*
ErrorEvent *Method*
Event *Method*
KeyEvent *Method*
MenuEvent *Method*
MouseEvent *Method*
MoveEvent *Method*
StatusEvent *Method*
TimerEvent *Method*
ValueEvent *Method*
Reports the status of an error flag.
errorCode ( ) LongInt

System *Procedure*
Returns a number describing the most recent run-time error or error condition.
errorCode ( ) SmallInt

errorLog

System *Procedure*
Adds information to the error stack.
errorLog (const ***errorCode*** SmallInt, const ***errorMessage*** String)

errorMessage

System *Procedure*
Returns the text of the most recent error message.
errorMessage ( ) String

errorPop

System *Procedure*
Removes an error from the top of the error stack.
errorPop ( ) Logical

errorShow

System *Procedure*
Displays the error dialog box.
errorShow ([const ***topHelp*** String
[, const ***bottomHelp*** String]]) Logical

errorTrapOnWarnings

System *Procedure*
Specifies whether to handle warning errors as critical errors.
errorTrapOnWarnings (const ***yesNo*** Logical)

exchange

Array *Method*
Swaps the contents of two cells in an array.
exchange (const ***index1*** LongInt, const ***index2*** LongInt)

execMethod

Library *Method*
Calls a custom method that takes no arguments.
execMethod (const ***methodName*** String)

UIObject *Method/Procedure*
Calls a custom method that takes no arguments.
execMethod (const ***methodName*** String)

execute

DDE *Method*
Sends a command via a DDE link.
execute (const ***command*** String) Logical

System *Procedure*
Executes a DOS command.
execute (const ***programName*** String [, const ***wait*** Logical
[, const ***displayMode*** SmallInt]]) Logical

executeQBE

Database *Method/Procedure*
Executes a QBE query.

1. **executeQBE** (const ***qbeVar*** Query) Logical
2. **executeQBE** (const ***qbeVar*** Query, const ***ansTbl*** String) Logical
3. **executeQBE** (const ***qbeVar*** Query, const ***ansTbl*** Table) Logical
4. **executeQBE** (const ***qbeVar*** Query, var ***ansTbl*** TCursor) Logical

Query *Method*
Executes a QBE query.

1. **executeQBE** ( ) Logical
2. **executeQBE** (const ***ansTbl*** String) Logical
3. **executeQBE** (const ***ansTbl*** Table) Logical
4. **executeQBE** (var ***ansTbl*** TCursor) Logical

executeQBEFile

Database *Method/Procedure*
Opens and executes a QBE file.

1. **executeQBEFile** (const ***qbeFileName*** String) Logical
2. **executeQBEFile** (const ***qbeFileName*** String, const ***ansTbl*** String) Logical
3. **executeQBEFile** (const ***qbeFileName*** String, const ***ansTbl*** Table) Logical
4. **executeQBEFile** (const ***qbeFileName*** String, var ***ansTbl*** TCursor) Logical

executeQBEString

Database *Method/Procedure*
Executes a QBE string.

1. **executeQBEString** (const ***QBEString*** String) Logical
2. **executeQBEString** (const ***QBEString*** String, const ***ansTbl*** String) Logical
3. **executeQBEString** (const ***QBEString*** String, const ***ansTbl*** Table) Logical
4. **executeQBEString** (const ***QBEString*** String, var ***ansTbl*** TCursor) Logical

existDrive

FileSystem *Method*
Reports whether a drive is attached to the system.
existDrive (const ***driveLetter*** String) Logical

exit

System *Procedure*
Closes Paradox and exits to Windows.
exit ( )

exp

Number *Method*
Returns the exponential (base *e*) of a number.
exp ( ) Number

fail

System *Procedure*
Causes a method to fail.
fail ([const ***errorNumber*** SmallInt, const ***errorMessage*** String])

familyRights

Table *Method*
Tests for a user's ability to create or modify objects in a table's family.
familyRights (const ***rights*** String) Logical

Compatibility Procedure
familyRights (const ***tableName*** String, const ***rights*** String) Logical

TCursor *Method*
Tests for a user's ability to create or modify objects in a table's family.
familyRights (const ***rights*** String) Logical

fieldName

Table *Method*
Returns the name of field in a table, given a field number.
fieldName (const ***fieldNum*** SmallInt) String

Compatibility Procedure
fieldName (const ***tableName*** String, const ***fieldNum*** SmallInt) String

TCursor *Method*
Returns the name of a field.
fieldName (const ***fieldNum*** SmallInt) String

fieldNo

Table *Method*
Returns the position of a field in a table.
fieldNo (const ***fieldName*** String) SmallInt

Compatibility Procedure
fieldNo (const ***tableName*** String, const ***fieldName*** String) SmallInt

TCursor *Method*
Returns the position of a field in a table.
fieldNo (const ***fieldName*** String) SmallInt

fieldRights

TCursor *Method*
Reports whether a user has rights to read or modify a field in a table.
1. fieldRights (const ***fieldName*** String, const ***rights*** String) Logical

2. fieldRights (const ***fieldNum*** SmallInt, const ***rights*** String) Logical

fieldSize

TCursor *Method*
Returns the size of a field.
1. fieldSize (const ***fieldName*** String) SmallInt

2. fieldSize (const ***fieldNum*** SmallInt) SmallInt

fieldType

Table *Method*
Returns the type of a field in a table.
1. fieldType (const ***fieldName*** String) String

2. fieldType (const ***fieldNum*** SmallInt) String

Compatibility Procedure
1. fieldType (const ***tableName*** String, const ***fieldName*** String) String

2. fieldType (const ***tableName*** String, const ***fieldNum*** SmallInt) String

TCursor *Method*
Returns the data type of a field.
1. fieldType (const ***fieldName*** String) String

2. fieldType (const ***fieldNum*** SmallInt) String

fieldUnits2

TCursor *Method*

Returns the number of decimal places defined for a numeric field in a dBASE table.

1. **fieldUnits2** (const ***fieldName*** String) SmallInt
2. **fieldUnits2** (const ***fieldNum*** SmallInt) SmallInt

fieldValue

TCursor *Method*

Reads the value of a specified field.

1. **fieldValue** (const ***fieldName*** String, var ***result*** AnyType) Logical
2. **fieldValue** (const ***fieldNum*** SmallInt, var ***result*** AnyType) Logical

fileBrowser

System *Procedure*

Displays the Browser and returns the names of one or more files selected by the user.

1. **fileBrowser** (var ***selectedFile*** String
   [, var ***browserInfo*** FileBrowserInfo]) Logical
2. **fileBrowser** (var ***selectedFiles*** Array[] String
   [, var ***browserInfo*** FileBrowserInfo]) Logical

fill

Array *Method*

Fills an array with a value.

fill (const ***value*** AnyType)

String *Procedure*

Returns a string containing repeated instances of a character.

fill (const ***fillCharacter*** String, const ***fillNumber*** SmallInt) String

findFirst

FileSystem *Method*

Searches a file system for a file name.

findFirst (const ***pattern*** String) Logical

findNext

FileSystem *Method*

Searches a file system for multiple instances of a file name.

findNext ([const ***fileSpec*** String]) Logical

floor

Number *Method*
Rounds a numeric expression down to the nearest whole number.
floor ( ) Number

format

String *Procedure*
Returns a formatted string for display or printing.
format (const ***formatSpec*** String, const ***value*** AnyType) String

formatAdd

System *Procedure*
Adds a format.
formatAdd (const ***formatName*** String, const ***formatSpec*** String) Logical

formatDelete

System *Procedure*
Deletes a format.
formatDelete (const ***formatName*** String) Logical

formatExist

System *Procedure*
Reports whether a format exists.
formatExist (const ***formatName*** String) Logical

formatSetCurrencyDefault

System *Procedure*
Sets the default display format for Currency values.
formatSetCurrencyDefault (const ***formatName*** String) Logical

formatSetDateDefault

System *Procedure*
Sets the default display format for Date values.
formatSetDateDefault (const ***formatName*** String) Logical

formatSetDateTimeDefault

System *Procedure*
Sets the default display format for DateTime values.
formatSetDateTimeDefault (const ***formatName*** String) Logical

formatSetLogicalDefault

System *Procedure*
Sets the default display format for Logical values.
formatSetLogicalDefault (const ***formatName*** String) Logical

formatSetLongIntDefault

System *Procedure*
Sets the default display format for LongInt values.
formatSetLongIntDefault (const ***formatName*** String) Logical

formatSetNumberDefault

System *Procedure*
Sets the default display format for Number values.
formatSetNumberDefault (const ***formatName*** String) Logical

formatSetSmallIntDefault

System *Procedure*
Sets the default display format for SmallInt values.
formatSetSmallIntDefault (const ***formatName*** String) Logical

formatSetTimeDefault

System *Procedure*
Sets the default display format for Time values.
formatSetTimeDefault (const ***formatName*** String) Logical

formCaller

Form *Procedure*
Creates a handle to the calling form.
formCaller (var ***caller*** Form) Logical

formReturn

Form *Procedure*
Returns control to a suspended method.
formReturn ([const ***returnValue*** AnyType])

fraction

Number *Method*
Returns the fractional part of a number.
fraction ( ) Number

freeDiskSpace

FileSystem *Method*
Returns the amount of free space on a drive.
freeDiskSpace (const ***driveLetter*** String) LongInt

fullName

FileSystem *Method*
Returns the full path of a file.
fullName ( ) String

fv

Number *Method*

Returns the future value of a series of equal payments.

fv (const ***interestRate*** Number, const ***periods*** Number) Number

getAliasPath

Session *Method/Procedure*

Returns the path for a specified alias.

getAliasPath (const ***aliasName*** String) String

getBoundingBox

UIObject *Method*

Returns the coordinates of the frame that bounds an object.

getBoundingBox (var ***topLeft*** Point, var ***bottomRight*** Point)

getDestination

MoveEvent *Method*

Reports which object is the destination of a move.

getDestination (var ***dest*** UIObject)

getDir

FileSystem *Method*

Returns the directory path pointed to by the FileSystem variable.

getDir ( ) String

getDrive

FileSystem *Method*

Returns the drive letter pointed to by the FileSystem variable.

getDrive ( ) String

getFileAccessRights

FileSystem *Procedure*

Reports access rights (also called file attributes) of a file.

getFileAccessRights (const ***fileName*** String) String

getKeys

DynArray *Method*

Loads a resizeable array with indexes of an existing DynArray.

getKeys (var ***keyNames*** Array[] String)

getLanguageDriver

TCursor *Method*

Returns the name of the current language driver for a table.

getLanguageDriver ( ) String

getLanguageDriverDesc

TCursor *Method*

Returns the description of the current language driver for a table.

getLanguageDriverDesc ( ) String

getMenuChoiceAttribute

Menu *Procedure*

Returns the display attributes of a menu item.

getMenuChoiceAttribute (const ***menuChoice*** String) SmallInt

getMenuChoiceAttributeById

Menu *Procedure*

Returns the display attribute of a menu item specified by its menu ID.

getMenuChoiceAttributeById (const ***menuId*** SmallInt) SmallInt

getMousePosition

MouseEvent *Method*

Reports on the mouse position relative to a container.

1. **getMousePosition** (var ***p*** Point)
2. **getMousePosition** (var ***xPosition*** LongInt, var ***yPosition*** LongInt)

getMouseScreenPosition

System *Procedure*

Returns the mouse position relative to the screen.

getMouseScreenPosition ( ) Point

getNetUserName

Session *Method/Procedure*

Returns the network user name for a session.

getNetUserName ( ) String

getObjectHit

MouseEvent *Method*

Creates a handle to the UIObject that received the event.

getObjectHit (var ***target*** UIObject) Logical

getPosition

Application *Method*

Form *Method/Procedure*

Report *Method*

TableView *Method*

Reports the position of a window onscreen.

getPosition (var ***x*** LongInt, var ***y*** LongInt, var ***w*** LongInt, var ***h*** LongInt)

UIObject *Method*

Locates the position of an object.

getPosition (const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt, const ***h*** LongInt)

getProperty

UIObject *Method*

Returns the value of a specified property.

getProperty (const ***propertyName*** String) AnyType

getPropertyAsString

UIObject *Method*

Returns the value of a specified property as a string.

getPropertyAsString (const ***propertyName*** String) String

getRGB

UIObject *Method*

Finds the red, green, and blue components of a color.

getRGB (const ***rgb*** LongInt, var ***red*** SmallInt, var ***green*** SmallInt, var ***blue*** SmallInt)

getServerName

OLE *Method*

Reports the name of the OLE server for an OLE object.

getServerName ( ) String

getTarget

ActionEvent *Method*
ErrorEvent *Method*
Event *Method*
KeyEvent *Method*
MenuEvent *Method*
MouseEvent *Method*
MoveEvent *Method*
StatusEvent *Method*
TimerEvent *Method*
ValueEvent *Method*

Creates a handle to the target of an Event.
getTarget (var ***target*** UIObject)

getTitle

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*

Returns the text of the window title bar.
getTitle ( ) String

getValidFileExtensions

FileSystem *Procedure*

Returns the valid file extensions for a specified object.
getValidFileExtensions (const ***objectType*** String) String

grow

Array *Method*

Increases the size of a resizeable array.
grow (const ***increment*** LongInt)

hasMenuChoiceAttribute

Menu *Procedure*

Reports whether a menu item contains a given display attribute.
hasMenuChoiceAttribute (const ***attrib*** SmallInt ,
const ***attribSet*** SmallInt) Logical

hasMouse

UIObject *Method*

Reports if the mouse is positioned over an object.
hasMouse ( ) Logical

helpOnHelp

System *Procedure*
Displays information about how to use the Help system.
helpOnHelp ( ) Logical

helpQuit

System *Procedure*
Tells the Help application it is no longer needed.
helpQuit (const ***helpFileName*** String) Logical

helpSetIndex

System *Procedure*
Sets the help index.
helpSetIndex (const ***helpFileName*** String, const ***indexId*** LongInt) Logical

helpShowContext

System *Procedure*
Displays help for a particular context.
helpShowContext (const ***helpFileName*** String,
const ***helpId*** LongInt) Logical

helpShowIndex

System *Procedure*
Displays the index of a specifed Help file.
helpShowIndex (const ***helpFileName*** String) Logical

helpShowTopic

System *Procedure*
Displays help for a specified topic.
helpShowTopic (const ***helpFileName*** String,
const ***topicKey*** String) Logical

helpShowTopicInKeywordTable

System *Procedure*
Displays help for a topic identified by a keyword in an alternate keyword table.
helpShowTopicInKeywordTable (const ***helpFileName*** String,
const ***keyTableLetter*** String, const ***topicKey*** String) Logical

hide

Application *Method*

Form *Method/Procedure*

Report *Method*

TableView *Method*

Makes a window invisible.

hide ( )

hideSpeedBar

Form *Procedure*

Makes the SpeedBar invisible.

hideSpeedBar ( )

home

TCursor *Method*

Moves to the first record of a table.

home ( ) Logical

TextStream *Method*

Sets the pointer to the beginning of a text file.

home( )

UIObject *Method*

Moves to the first record of a table.

home ( ) Logical

hour

DateTime *Method*

Time *Method*

Extracts as a number the hour from a DateTime.

hour ( ) SmallInt

id

ActionEvent *Method*

Returns the ID number of an ActionEvent.

id ( ) SmallInt

MenuEvent *Method*

Returns the ID number of a MenuEvent.

id ( ) SmallInt

ignoreCaseInLocate

Session *Procedure*

Specifies whether to ignore case in locate operations.

ignoreCaseInLocate ([const ***yesNo*** Logical])

ignoreCaseInStringCompares

String *Procedure*

Specifies whether to consider case when comparing strings.

ignoreCaseInStringCompares (const ***yesNo*** Logical)

index

Table *Keyword*

Creates a primary or secondary index on the specified fields of a table.

1. index

[maintained] *tableDesc* **on** *fieldID*

endIndex

2. index *tableDesc*

[maintained] (Paradox)

[descending] (dBASE)

[unique] (dBASE)

[primary] (Paradox)

[caseInsensitive] (Paradox)

on

{ *fieldDesc* [, *fieldDesc*] **[to** *indexName*]

|

{ *keyExp*

to *ndxFileName* | **tag** *tagName* [**of** *mdxFileName*]

|

for *condition* } }

endIndex

indexOf

Array *Method*

Returns the position of an item in an array.

indexOf (const ***value*** AnyType) LongInt

initRecord

TCursor *Method*

Empties the record buffer.

initRecord ( ) Logical

insert

Array *Method*
Inserts one or more empty cells into an array.
insert (const ***Index*** LongInt [, const ***numberOfItems*** LongInt])

insertAfter

Array *Method*
Inserts an item into an array after a specified item.
insertAfter (const ***keyItem*** AnyType, const ***insertedItem*** AnyType)

insertAfterRecord

TCursor *Method*
Inserts a record into a table after the current record.
insertAfterRecord ([const ***pointer*** TCursor]) Logical

UIObject *Method*
Inserts a record into a table after the current record.
insertAfterRecord ( ) Logical

insertBefore

Array *Method*
Inserts an item into an array before a specified item.
insertBefore (const ***keyItem*** AnyType, const ***insertedItem*** AnyType)

insertBeforeRecord

TCursor *Method*
Inserts a record into a table before the current record.
insertBeforeRecord ([const ***pointer*** TCursor]) Logical

UIObject *Method*
Inserts a record into a table before the current record.
insertBeforeRecord ( ) Logical

insertFirst

Array *Method*
Inserts an item at the beginning of an array.
insertFirst (const ***value*** AnyType)

insertRecord

TCursor *Method*
Inserts a record into a table.
insertRecord ([const ***pointer*** TCursor]) Logical

UIObject *Method*
Inserts a record into a table.
insertRecord ( ) Logical

int

SmallInt *Procedure*
Casts a value as an integer.
int (const ***value*** AnyType) SmallInt

isAbove

Point *Method*
Reports whether a point is above another point.
isAbove (const ***pt*** Point) Logical

isAdvancedWildcardsInLocate

Session *Procedure*
Reports whether this session is using advanced wildcards in locate operations.
isAdvancedWildcardsInLocate ( ) Logical

isAltKeyDown

KeyEvent *Method*
Reports whether *Alt* was held down during a KeyEvent.
isAltKeyDown ( ) Logical

isAssigned

AnyType *Method*
Reports whether a variable has been assigned a value.
isAssigned ( ) Logical

Database *Method*
Reports whether a Database variable has been assigned a value.
isAssigned ( ) Logical

Query *Method*
Reports whether a Query variable has been assigned a value.
isAssigned( ) Logical

Session *Method*
Reports whether a Session variable has been assigned a value.
isAssigned ( ) Logical

Table *Method*
Reports whether a Table variable has been assigned a value.
isAssigned ( ) Logical

TCursor *Method*
Reports whether a TCursor variable has been assigned a value.
isAssigned ( ) Logical

isBelow

Point *Method*
Reports whether a point is below another point.
isBelow (const ***pt*** Point) Logical

isBlank

AnyType *Method*
Reports whether an expression has a blank value.
isBlank ( ) Logical

isBlankZero

Session *Method/Procedure*
Reports whether blank values are being treated as zeros in calculations.
isBlankZero ( ) Logical

isContainerValid

UIObject *Procedure*
Reports whether an object's container is valid.
isContainerValid ( ) Logical

isControlKeyDown

KeyEvent *Method*
Reports whether *Ctrl* was held down during a KeyEvent.
isControlKeyDown ( ) Logical

MouseEvent *Method*
Reports whether *Ctrl* was held down during a MouseEvent.
isControlKeyDown ( ) Logical

isDir

FileSystem *Procedure*
Reports whether a specified string represents the name of a directory.
isDir (const ***dirName*** String) Logical

isEdit

TCursor *Method*
Reports whether a TCursor is in Edit mode.
isEdit ( ) Logical

UIObject *Method*
Reports whether an object is in Edit mode.
isEdit ( ) Logical

isEmpty

Table *Method*
Reports whether a table contains any records.
isEmpty ( ) Logical

Compatibility Procedure
isEmpty (const ***tableName*** String) Logical

TCursor *Method*
Reports whether a table contains any records.
isEmpty ( ) Logical

UIObject *Method*
Reports whether a table contains any records.
isEmpty ( ) Logical

isEncrypted

Table *Method*
Reports whether a table is encrypted.
isEncrypted ( ) Logical

Compatibility Procedure
isEncrypted (const ***tableName*** String) Logical

TCursor *Method*
Reports whether a table is encrypted.
isEncrypted ( ) Logical

isFile

FileSystem *Procedure*
Reports whether a specified string is the name of a file in the current file system.
isFile (const ***fileName*** String) Logical

isFirstTime

ActionEvent *Method*
ErrorEvent *Method*
Event *Method*
KeyEvent *Method*
MenuEvent *Method*
MouseEvent *Method*
MoveEvent *Method*
StatusEvent *Method*
TimerEvent *Method*
ValueEvent *Method*

Reports whether the form is handling an Event for the first time before dispatching it.

isFirstTime ( ) Logical

isFixed

FileSystem *Method*

Reports whether a drive is fixed.

isFixed (const ***driveLetter*** String) Logical

isFixedType

AnyType *Method*

Reports whether a variable's data type has been explicitly declared.

isFixedType ( ) Logical

isFromUI

KeyEvent *Method*

Reports whether an event was generated by the user interacting with Paradox.

isFromUI ( ) Logical

MenuEvent *Method*

Reports whether an event was generated by the user interacting with Paradox.

isFromUI ( ) Logical

MouseEvent *Method*

Reports whether an event was generated by the user interacting with Paradox.

isFromUI ( ) Logical

isIgnoreCaseInLocate

Session *Procedure*
Reports whether the current session is ignoring case in locate operations.
isIgnoreCaseInLocate ( ) Logical

isIgnoreCaseInStringCompares

String *Procedure*
Reports whether case is considered when comparing strings.
isIgnoreCaseInStringCompares ( ) Logical

isInside

MouseEvent *Method*
Reports whether the mouse is inside the border of the target object.
isInside ( ) Logical

isLastMouseClickedValid

UIObject *Procedure*
Reports if the last object to receive a mouse click is valid.
isLastMouseClickedValid ( ) Logical

isLastMouseRightClickedValid

UIObject *Procedure*
Reports if the last object to receive a right mouse click is valid.
isLastMouseRightClickedValid ( ) Logical

isLeapYear

Date *Method*
DateTime *Method*
Reports whether a year has 366 days.
isLeapYear ( ) Logical

isLeft

Point *Method*
Reports whether a point is to the left of another point.
isLeft (const ***pt*** Point) Logical

isLeftDown

MouseEvent *Method*
Reports whether the left (or primary) mouse button is held down during a MouseEvent.
isLeftDown ( ) Logical

isMaximized

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*
Reports whether a window is displayed at its maximum size.
isMaximized ( ) Logical

isMiddleDown

MouseEvent *Method*
Reports whether the middle mouse button is held down during a MouseEvent.
isMiddleDown ( ) Logical

isMinimized

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*
Reports whether a window is displayed as an icon.
isMinimized ( ) Logical

isPreFilter

ActionEvent *Method*
ErrorEvent *Method*
Event *Method*
KeyEvent *Method*
MenuEvent *Method*
MouseEvent *Method*
MoveEvent *Method*
StatusEvent *Method*
TimerEvent *Method*
ValueEvent *Method*
Reports whether the form is handling an Event on its own behalf.
isPreFilter ( ) Logical

isRecordDeleted

TCursor *Method*
Reports whether the current record has been deleted (dBASE tables only).
isRecordDeleted ( ) Logical

UIObject *Method*
Reports whether the current record has been deleted (dBASE tables only).
isRecordDeleted ( ) Logical

isRemote

FileSystem *Method*
Reports whether a drive is remote (a network drive).
isRemote (const ***driveLetter*** String) Logical

isRemovable

FileSystem *Method*
Reports whether a drive is removable.
isRemovable (const ***driveLetter*** String) Logical

isResizeable

Array *Method*
Reports whether an array can be resized.
isResizeable ( ) Logical

isRight

Point *Method*
Reports whether a point is to the right of another point.
isRight (const ***pt*** Point) Logical

isRightDown

MouseEvent *Method*
Reports whether the right mouse button is held down during a MouseEvent.
isRightDown ( ) Logical

isShared

Table *Method*
Reports whether a table is currently shared.
isShared ( ) Logical

Compatibility Procedure
isShared (const ***tableName*** String) Logical

TCursor *Method*
Reports whether a table is currently shared.
isShared ( ) Logical

isShiftKeyDown

KeyEvent *Method*
Reports whether ***Shift*** was held down during a KeyEvent.
isShiftKeyDown ( ) Logical

MouseEvent *Method*
Reports whether ***Shift*** was held down during a MouseEvent.
isShiftKeyDown ( ) Logical

isShowDeletedOn

TCursor *Method*
Reports whether deleted records in a dBASE table are shown.
isShowDeletedOn ( ) Logical

isSpace

String *Method*
Reports whether a string contains only spaces (or is empty).
isSpace (const ***string*** String) Logical

isSpeedBarShowing

Form *Procedure*
Reports whether the SpeedBar is visible.
isSpeedBarShowing ( ) Logical

isTable

Database *Method/Procedure*
Reports whether a table exists in a database.
1. isTable (const ***tableName*** String [, const ***tableType*** String]) Logical

2. isTable (const ***tableVar*** Table) Logical

Table *Method*
Reports whether a table exists in a database.
isTable ( ) Logical

Compatibility Procedure
isTable (const ***tableName*** String) Logical

isTargetSelf

ActionEvent *Method*
ErrorEvent *Method*
Event *Method*
KeyEvent *Method*
MenuEvent *Method*
MouseEvent *Method*
MoveEvent *Method*
StatusEvent *Method*
TimerEvent *Method*
ValueEvent *Method*

Reports whether an object is the target of an Event.
isTargetSelf ( ) Logical

isValid

TCursor *Method*

Reports whether the contents of a field are legitimate and complete.
1. **isValid** (const ***fieldName*** String, const ***value*** AnyType) Logical
2. **isValid** (const ***fieldNum*** SmallInt, const ***value*** AnyType) Logical

isVisible

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*

Reports whether any part of a window is displayed.
isVisible ( ) Logical

keyChar

Form *Method*

Sends an event to a form's **keyChar** method.
1. **keyChar** (const ***aChar*** SmallInt, const ***vChar*** SmallInt, const ***state*** SmallInt) Logical
2. **keyChar** (const ***characters*** String [, const ***state*** SmallInt]) Logical

UIObject *Method*

Sends an event to an object's **keyChar** method.
1. **keyChar** (const ***characters*** String [, const ***state*** SmallInt]) Logical
2. **keyChar** (const ***ansiKeyValue*** SmallInt) Logical
3. **keyChar** (const ***ansiKeyValue*** SmallInt, const ***vChar*** SmallInt, const ***state*** SmallInt) Logical

keyNameToChr

String *Procedure*
Returns the one-character string represented by a virtual key-code string.
keyNameToChr (const ***keyName*** String) String

keyNameToVKCode

String *Procedure*
Returns the numeric virtual key code of a virtual key-code string.
keyNameToVKCode (const ***keyName*** String) SmallInt

keyPhysical

Form *Method*
Sends an event to a form's **keyPhysical** method.
keyPhysical (const ***aChar*** SmallInt, const ***vChar*** SmallInt, const ***state*** SmallInt) Logical

UIObject *Method*
Sends an event to an object's **keyPhysical** method.
keyPhysical (const ***ansiKeyValue*** SmallInt, const ***vChar*** SmallInt, const ***state*** SmallInt) Logical

killTimer

UIObject *Method*
Stops the timer associated with an object.
killTimer ( )

ln

Number *Method*
Returns the natural logarithm of a numeric expression.
ln ( ) Number

load

Form *Method*
Opens a form in the Form Design window.
load (const ***formName*** String) Logical

Report *Method*
Opens a report in the Report Design window.
load (const ***reportName*** String) Logical

locate

TCursor *Method*

Searches for a specified field value.

1. **locate** (const ***fieldName*** String, const ***exactMatch*** AnyType
   [, const ***fieldName*** String, const ***exactMatch*** AnyType]*) Logical

2. **locate** (const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType
   [, const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType]*) Logical

UIObject *Method*

Searches for a specified value.

1. **locate** (const ***fieldName*** String, const ***exactMatch*** AnyType
   [, const ***fieldName*** String, const ***exactMatch*** AnyType]*) Logical

2. **locate** (const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType
   [, const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType]*) Logical

locateNext

TCursor *Method*

Searches for a specified field value.

1. **locateNext** (const ***fieldName*** String, const ***exactMatch*** AnyType
   [, const ***fieldName*** String, const ***exactMatch*** AnyType]*) Logical

2. **locateNext** (const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType
   [, const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType]*) Logical

UIObject *Method*

Searches forward from the current record for a specified field value.

1. **locateNext** (const ***fieldName*** String, const ***exactMatch*** AnyType
   [,const ***fieldName*** String, const ***exactMatch*** AnyType]*) Logical

2. **locateNext** (const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType
   [,const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType]*) Logical

locateNextPattern

TCursor *Method*

Locates the next record containing a field that has a specified pattern of characters.

1. **locateNextPattern** ([const ***fieldName*** String,
   const ***exactMatch*** AnyType,] * const ***fieldName*** String,
   const ***pattern*** String) Logical

2. **locateNextPattern** ([const ***fieldNum*** SmallInt,
   const ***exactMatch*** AnyType,] * const ***fieldNum*** SmallInt,
   const ***pattern*** String) Logical

UIObject *Method*

Locates the next record containing a field that has a specified pattern of characters.

1. **locateNextPattern** ([const ***fieldName*** String, const ***exactMatch*** AnyType,] * const ***fieldName*** String, const ***pattern*** String) Logical
2. **locateNextPattern** ([const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType,] * const ***fieldNum*** SmallInt, const ***pattern*** String) Logical

locatePattern

TCursor *Method*

Locates a record containing a field that has a specified pattern of characters.

1. **locatePattern** ([const ***fieldName*** String, const ***exactMatch*** AnyType,]* const ***fieldName*** String, const ***pattern*** String) Logical
2. **locatePattern** ([const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType,]* const ***fieldNum*** SmallInt, const ***pattern*** String) Logical

UIObject *Method*

Searches for a record containing a field that has a specified pattern of characters.

1. **locatePattern** ([const ***fieldName*** String, const ***exactMatch*** AnyType,]* const ***fieldName*** String, const ***pattern*** String) Logical
2. **locatePattern** ([const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType,]* const ***fieldNum*** SmallInt, const ***pattern*** String) Logical

locatePrior

TCursor *Method*

Searches for a specified field value.

1. **locatePrior** (const ***fieldName*** String, const ***exactMatch*** AnyType [, const ***fieldName*** String, const ***exactMatch*** AnyType]*) Logical
2. **locatePrior** (const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType [, const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType]*) Logical

UIObject *Method*

Searches backward for a specified field value.

1. **locatePrior** (const ***fieldName*** String, const ***exactMatch*** AnyType [, const ***fieldName*** String, const ***exactMatch*** AnyType]*) Logical
2. **locatePrior** (const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType [, const ***fieldNum*** SmallInt, const ***exactMatch*** AnyType]*) Logical

locatePriorPattern

TCursor *Method*

Locates the previous record containing a field that has a specified pattern of characters.

1. **locatePriorPattern** ([const ***fieldName*** String,
   const ***exactMatch*** AnyType,] * const ***fieldName*** String,
   const ***pattern*** String) Logical

2. **locatePriorPattern** ([const ***fieldNum*** SmallInt,
   const ***exactMatch*** AnyType,] * const ***fieldNum*** SmallInt,
   const ***pattern*** String) Logical

UIObject *Method*

Locates the prior record containing a field that has a specified pattern of characters.

1. **locatePriorPattern** ([const ***fieldName*** String,
   const ***exactMatch*** AnyType,] * const ***fieldName*** String,
   const ***pattern*** String) Logical

2. **locatePriorPattern** ([const ***fieldNum*** SmallInt,
   const ***exactMatch*** AnyType,] * const ***fieldNum*** SmallInt,
   const ***pattern*** String) Logical

lock

Session *Procedure*

Locks one or more tables.

lock (const ***table*** { Table | TCursor | String}, const ***lockType*** String
[, const ***table*** { Table | TCursor | String },
const ***lockType*** String]*) Logical

Table *Method*

Locks a specified table.

lock (const ***lockType*** String) Logical

TCursor *Method*

Locks a specified table.

lock (const ***lockType*** String) Logical

lockRecord

TCursor *Method*

Puts a write lock on the current record.

lockRecord ( ) Logical

UIObject *Method*

Puts a write lock on the current record.

lockRecord ( ) Logical

lockStatus

TCursor *Method*
Returns the number of times you have placed a lock on a table.
lockStatus (const ***lockType*** String) SmallInt

UIObject *Method*
Returns the number of times you have placed a lock on a table.
lockStatus (const ***lockType*** String) SmallInt

log

Number *Method*
Returns the base 10 logarithm of a numeric expression.
log ( ) Number

logical

Logical *Procedure*
Casts a value as type Logical.
logical (const ***value*** AnyType) Logical

longInt

LongInt *Procedure*
Casts a value as a LongInt.
longInt (const ***value*** AnyType) LongInt

lower

String *Method*
Converts a string to lowercase.
lower ( ) String

lTrim

String *Method*
Removes leading blanks from a string.
lTrim ( ) String

makeDir

FileSystem *Method*
Creates a new directory.
makeDir (const ***name*** String) Logical

match

String *Method*
Compares a string with a pattern.
match (const ***pattern*** String [, var ***matchVar*** String]*) Logical

max

Number *Procedure*
Returns the larger of two numbers.
max (const ***x1*** AnyType, const ***x2*** AnyType) AnyType

maximize

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*
Maximizes a window.
maximize ( )

memo

Memo *Procedure*
Casts a value as a Memo.
memo (const ***value*** AnyType [, const ***value*** AnyType]*) String

menuAction

Form *Method/Procedure*
Sends an event to a form's **menuAction** method.
menuAction (const ***action*** SmallInt) Logical

UIObject *Method/Procedure*
Sends an event to an object's **menuAction** method.
menuAction (const ***action*** SmallInt) Logical

menuChoice

MenuEvent *Method*
Returns a string containing an item chosen from a menu.
menuChoice ( ) String

message

System *Procedure*
Displays a message in the status line.
message (const ***message*** String [, const ***message*** String]*)

methodDelete

Form *Method*
Deletes a form-level method from a form.
methodDelete (const ***methodName*** String) Logical

UIObject *Method*
Deletes a specified method.
methodDelete (const ***methodName*** String) Logical

methodGet

Form *Method*
Returns the text of a form-level method.
methodGet (const ***methodName*** String) String

UIObject *Method*
Returns the text of a specified method.
methodGet (const ***methodName*** String) String

methodSet

Form *Method*
Sets the definition of a form-level method.
methodSet (const ***methodName*** String,
const ***methodText*** String) Logical

UIObject *Method*
Sets the definition of a specified method.
methodSet (const ***methodName*** String, const ***methodText*** String) Logical

milliSec

DateTime *Method*

Time *Method*
Extracts as a number the milliseconds from a DateTime.
milliSec ( ) SmallInt

min

Number *Procedure*
Returns the smaller of two numbers.
min (const ***x1*** AnyType, const ***x2*** AnyType) AnyType

minimize

Application *Method*

Form *Method/Procedure*

Report *Method*

TableView *Method*
Minimizes a window.
minimize ( )

minute

DateTime *Method*

Time *Method*
Extracts as a number the minutes from a DateTime.
minute ( ) SmallInt

mod

Number *Method*

Returns the remainder when one number is divided by another.

mod (const ***modulo*** Number) Number

month

Date *Method*

DateTime *Method*

Extracts as a number the month from a DateTime.

month ( ) SmallInt

mouseClick

UIObject *Method*

Sends an event to an object's **mouseClick** method.

mouseClick ( ) Logical

mouseDouble

Form *Method*

Sends an event to a form's **mouseDouble** method.

mouseDouble (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

UIObject *Method*

Sends an event to an object's **mouseDouble** method.

mouseDouble (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

mouseDown

Form *Method*

Sends an event to a form's **mouseDown** method.

mouseDown (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

UIObject *Method*

Sends an event to an object's **mouseDown** method.

mouseDown (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

mouseEnter

Form *Method*
Sends an event to a form's **mouseEnter** method.
mouseEnter (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

UIObject *Method*
Sends an event to an object's **mouseEnter** method.
mouseEnter (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

mouseExit

Form *Method*
Sends an event to a form's **mouseExit** method.
mouseExit (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

UIObject *Method*
Sends an event to an object's **mouseExit** method.
mouseExit (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

mouseMove

Form *Method*
Sends an event to a form's **mouseMove** method.
mouseMove (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

UIObject *Method*
Sends an event to an object's **mouseMove** method.
mouseMove (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

mouseRightDouble

Form *Method*
Sends an event to a form's **mouseRightDouble** method.
mouseRightDouble (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

UIObject *Method*
Sends an event to an object's **mouseRightDouble** method.
mouseRightDouble (const ***x*** LongInt, const ***y*** LongInt,
const ***state*** SmallInt) Logical

mouseRightDown

Form *Method*

Sends an event to a form's **mouseRightDown** method.

mouseRightDown (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

UIObject *Method*

Sends an event to an object's **mouseRightDown** method.

mouseRightDown (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

mouseRightUp

Form *Method*

Sends an event to a form's **mouseRightUp** method.

mouseRightUp (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

UIObject *Method*

Sends an event to an object's **mouseRightUp** method.

mouseRightUp (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

mouseUp

Form *Method*

Sends an event to a form's **mouseUp** method.

mouseUp (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

UIObject *Method*

Sends an event to an object's **mouseUp** method.

mouseUp (const ***x*** LongInt, const ***y*** LongInt, const ***state*** SmallInt) Logical

moveTo

Form *Method*

Moves to a form.

moveTo ([const ***objectName*** String]) Logical

UIObject *Method*

Sets the focus to a specified object name.

moveTo ( ) Logical

Procedure

moveTo (const ***objectName*** String) Logical

moveToPage

Form *Method/Procedure*
Displays a specified page of a form.
moveToPage (const ***pageNumber*** SmallInt) Logical

Report *Method*
Displays a specified page of a report.
moveToPage (const ***pageNumber*** SmallInt) Logical

moveToRecNo

TCursor *Method*
Moves a TCursor to a specific record in a table.
moveToRecNo (const ***recordNum*** LongInt) Logical

UIObject *Method*
Moves to a specific record in a dBASE table.
moveToRecNo (const ***recordNum*** LongInt) Logical

moveToRecord

TableView *Method*
Moves to a specific record in a table.
moveToRecord (const ***tc*** TCursor) Logical

TCursor *Method*
Moves a TCursor to a specific record in a table.
moveToRecord (const ***recordNum*** LongInt) Logical

UIObject *Method*
Moves to a specific record in a table.
1. **moveToRecord** (const ***recordNum*** LongInt) Logical
2. **moveToRecord** (const ***tc*** TCursor) Logical

moy

Date *Method*

DateTime *Method*
Extracts as a string the month from a Date or DateTime.
moy ( ) String

msgAbortRetryIgnore

System *Procedure*
Displays a dialog box containing a message and three buttons: Abort, Retry, and Ignore.
msgAbortRetryIgnore (const ***caption*** String, const ***text*** String) String

msgInfo

System *Procedure*

Displays a dialog box containing the information icon, a message, and an OK button.

msgInfo (const ***caption*** String, const ***text*** String)

msgQuestion

System *Procedure*

Displays a dialog box containing a message, a question mark icon, a Yes button, and a No button.

msgQuestion (const ***caption*** String, const ***text*** String) String

msgRetryCancel

System *Procedure*

Displays a dialog box containing a message and two buttons: Retry and Cancel.

msgRetryCancel (const ***caption*** String, const ***text*** String) String

msgStop

System *Procedure*

Displays a dialog box containing a stop sign icon, a message, and an OK button.

msgStop (const ***caption*** String, const ***text*** String)

msgYesNoCancel

System *Procedure*

Displays a dialog box containing a message and three buttons: Yes, No, and Cancel.

msgYesNoCancel (const ***caption*** String, const ***text*** String) String

name

FileSystem *Method*

Returns the name of a file.

name ( ) String

newValue

ValueEvent *Method*

Returns the unposted new value of a ValueEvent.

newValue ( ) AnyType

nextRecord

TCursor *Method*
Moves to the next record in a table.
nextRecord ( ) Logical

UIObject *Method*
Moves to the next record in a table.
nextRecord ( ) Logical

nFields

Table *Method*
Returns the number of fields in a table.
nFields ( ) LongInt

Compatibility Procedure
nFields (const ***tableName*** String) LongInt

TCursor *Method*
Returns the number of fields in a table.
nFields ( ) LongInt

UIObject *Method*
Returns the number of fields in a table.
nFields ( ) LongInt

nKeyFields

Table *Method*
Returns the number of fields in the primary or current index for a table.
nKeyFields ( ) LongInt

Compatibility Procedure
nKeyFields (const ***tableName*** String) LongInt

TCursor *Method*
Returns the number of fields in the current index of a table.
nKeyFields ( ) LongInt

UIObject *Method*
Returns the number of key fields in a table.
nKeyFields ( ) LongInt

nRecords

Table *Method*
Returns the number of records in a table.
nRecords ( ) LongInt

Compatibility Procedure
nRecords (const ***tableName*** String) LongInt

TCursor *Method*
Returns the number of records in a table.
nRecords ( ) LongInt

UIObject *Method*
Returns the number of records in a table.
nRecords ( ) LongInt

number

Number *Procedure*
Casts a value as a Number.
number (const ***value*** AnyType) Number

numVal

Number *Procedure*
Casts a value as a Number.
numVal (const ***value*** AnyType) Number

oemCode

String *Procedure*
Returns the OEM code of a one-character string.
oemCode (const ***char*** String) SmallInt

open

Database *Method*
Opens a database.

1. **open** ( ) Logical
2. **open** (const ***databaseName*** String) Logical
3. **open** (const ***ses*** Session) Logical
4. **open** (const ***databaseName*** String, const ***ses*** Session) Logical

DDE *Method*

Opens a DDE link to another application.

1. **open** (const ***server*** String) Logical

2. **open** (const ***server*** String, const ***topic*** String) Logical

3. **open** (const ***server*** String, const ***topic*** String,
   const ***item*** String) Logical

Form *Method*

Opens a window.

1. **open** (const ***formName*** String
   [, const ***windowStyle*** LongInt]) Logical

2. **open** (const ***formName*** String, const ***windowStyle*** LongInt,
   const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt,
   const ***h*** LongInt) Logical

3. **open** (const ***openInfo*** FormOpenInfo) Logical

Library *Method*

Associates a Library variable with a library, and makes the library code available.

open (const ***libraryName*** String [, const ***libScope*** SmallInt]) Logical

Report *Method*

Opens a report and prints it.

1. **open** (const ***reportName*** String [, const ***windowStyle*** LongInt]) Logical

2. **open** (const ***reportName*** String, const ***windowStyle*** LongInt,
   const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt,
   const ***h*** LongInt) Logical

3. **open** (const ***openInfo*** ReportOpenInfo) Logical

Session *Method*

Opens a session.

open ([const ***sessionName*** String]) Logical

TableView *Method*

Opens a Table window.

1. **open** (const ***tvName*** String [, const ***windowStyle***
   LongInt]) Logical

2. open (const ***tvName*** String, const ***windowStyle*** LongInt,
   const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt,
   const ***h*** LongInt) Logical

TCursor *Method*

Opens a table.

1. **open** (const ***tableName*** String [, const ***db*** DataBase]
   [, const ***indexName*** String]) Logical

2. **open** (const ***tableVar*** Table) Logical

TextStream *Method*
Opens a text file in a specified mode.
open (const ***fileName*** String, const ***mode*** String) Logical

openAsDialog

Form *Method*
Opens a Form window as a dialog box.
1. **openAsDialog** (const ***formName*** String
   [, const ***windowStyle*** LongInt]) Logical
2. **openAsDialog** (const ***formName*** String, const ***windowStyle*** LongInt,
   const ***x*** LongInt, const ***y*** LongInt,
   const ***w*** LongInt, const ***h*** LongInt) Logical
3. **openAsDialog** (const ***openInfo*** FormOpenInfo) Logical

pixelsToTwips

System *Procedure*
Converts screen coordinates from pixels to twips.
pixelsToTwips (const ***pixels*** Point) Point

UIObject *Method*
Converts screen coordinates from pixels to twips.
pixelsToTwips (const ***pixels*** Point) Point

play

System *Procedure*
Plays a standalone script.
play (const ***scriptName*** String) AnyType

pmt

Number *Method*
Returns the periodic payment required to pay off a loan.
pmt (const ***interestRate*** Number, const ***periods*** Number) Number

point

Point *Procedure*
Casts an expression as a Point.
1. **point** ([const ***x*** LongInt, const ***y*** LongInt]) Point
2. **point** (const ***newPoint*** Point) Point

position

TextStream *Method*
Returns the pointer's position in a text file.
position ( ) LongInt

postAction

Form — *Method*
Posts an action to an action queue for delayed execution.
postAction (const ***actionId*** SmallInt)

UIObject — *Method*
Posts an action to an action queue for delayed execution.
postAction (const ***actionId*** SmallInt)

postRecord

TCursor — *Method*
Posts changes to a record.
postRecord ( ) Logical

UIObject — *Method*
Posts a pending record to a table.
postRecord ( ) Logical

pow

Number — *Method*
Raises a number to a power.
pow (const ***exponent*** Number) Number

pow10

Number — *Method*
Calculates 10 to a specified power.
pow10 ( ) Number

print

Report — *Method*
Prints a report.

1. **print** ( ) Logical
2. **print** (const ***reportName*** String,
   const ***reportPrintRestart*** SmallInt) Logical
3. **print** (const ***ri*** ReportPrintInfo) Logical

priorRecord

TCursor — *Method*
Moves to the previous record in a table.
priorRecord ( ) Logical

UIObject — *Method*
Moves to the previous record in a table.
priorRecord ( ) Logical

privDir

FileSystem *Procedure*

Returns the name of the user's private directory.

privDir ( ) String

protect

Table *Method*

Encrypts and assigns an owner password to a table.

protect ([const ***password*** String]) Logical

Compatibility Procedure

protect (const ***tableName*** String [, const ***password*** String]) Logical

pushButton

UIObject *Method*

Sends an event to an object's **pushButton** method.

pushButton ( ) Logical

pv

Number *Method*

Returns the present value of a series of equal payments.

pv (const ***interestRate*** Number, const ***periods*** Number) Number

qLocate

TCursor *Method*

Searches an indexed table for a specified field value.

qLocate (const ***searchValue*** AnyType
[, const ***searchValue*** AnyType]*) Logical

query

Query *Keyword*

Begins a query statement.

query

tableName | *fieldName* | [*fieldName* |]*
| *criteria* | [*criteria* |]*

[*tableName* | *fieldName* | [*fieldName* |]*
| *criteria* | [*criteria* |]*] *

endQuery

rand

Number *Procedure*

Generates a random value ranging from 0 to 1.

rand ( ) Number

readChars

TextStream *Method*
Reads a specified number of characters from a text file.
readChars (var ***string*** String, const ***nChars*** SmallInt) Logical

readEnvironmentString

System *Procedure*
Reads an item from the DOS environment.
readEnvironmentString (const ***key*** String) String

readFromClipboard

Graphic *Method*
Reads a graphic from the Clipboard.
readFromClipboard ( ) Logical

OLE *Method*
Pastes an OLE object from the Clipboard into an OLE variable.
readFromClipboard ( ) Logical

readFromFile

Binary *Method*
Reads data from a file and stores it in a Binary variable.
readFromFile (const ***fileName*** String) Logical

Graphic *Method*
Reads a graphic from a file.
readFromFile (const ***fileName*** String) Logical

Memo *Method*
Reads a memo from a file.
readFromFile (const ***fileName*** String) Logical

readLine

TextStream *Method*
Reads a line (or lines) from a text file.
1. readLine (var ***value*** String) Logical

2. readLine (var ***stringArray*** Array[] String) Logical

readProfileString

System *Procedure*
Reads an item from an initial settings file.
readProfileString (const ***fileName*** String, const ***section*** String, const ***key*** String) String

reason

ActionEvent *Method*

Event *Method*

KeyEvent *Method*

MouseEvent *Method*

TimerEvent *Method*

ValueEvent *Method*
Reports why an Event occurred.
reason ( ) SmallInt

ErrorEvent *Method*
Reports why an error occurred.
reason ( ) SmallInt

MenuEvent *Method*
Reports the type of menu chosen.
reason ( ) SmallInt

MoveEvent *Method*
Reports why a move occurred.
reason ( ) SmallInt

StatusEvent *Method*
Reports where a status message occurred.
reason ( ) SmallInt

recNo

TCursor *Method*
Returns the record number of the current record.
recNo ( ) LongInt

recordStatus

TCursor *Method*
Reports about the status of a record.
recordStatus (const ***statusType*** String) Logical

UIObject *Method*
Reports about the status of a record.
recordStatus (const ***statusType*** String) Logical

reIndex

Table *Method*
Rebuilds specified index files.
1. (Paradox tables) **reIndex** (const ***indexName*** String) Logical

2. (dBASE tables) **reIndex** (const ***indexName*** String
[, const ***tagName*** String]) Logical

TCursor *Method*
Rebuilds specified index files.
reIndex (const ***IndexName*** String [, const ***TagName*** String]) Logical

reIndexAll

Table *Method*
Rebuilds all index files associated with a table.
reIndexAll ( ) Logical

TCursor *Method*
Rebuilds all index files for a table.
reIndexAll ( ) Logical

remove

Array *Method*
Removes one or more items from an array.
remove (const ***index*** SmallInt [, const ***numberOfItems*** SmallInt])

Menu *Method*

PopUpMenu *Method*
Removes an item from a menu.
remove (const ***item*** String)

removeAlias

Session *Method/Procedure*
Removes an alias from a session.
removeAlias (const ***aliasName*** String) Logical

removeAllItems

Array *Method*
Removes all occurrences of an array item.
removeAllItems (const ***value*** AnyType)

removeAllPasswords

Session *Method/Procedure*
Removes all passwords presented to a session.
removeAllPasswords ( )

removeItem

Array *Method*
Deletes a specified item from an Array.
removeItem (const ***value*** AnyType)

DynArray *Method*
Deletes a specified item from a DynArray.
removeItem (const ***value*** AnyType)

removeMenu

Menu *Procedure*
Removes a custom menu and restores the default menu.
removeMenu ( )

removePassword

Session *Method/Procedure*
Removes a password presented to a session.
removePassword (const ***password*** String)

rename

FileSystem *Method*
Renames a file.
rename (const ***oldName*** String, const ***newName*** String) Logical

Table *Method*
Renames a table.
rename (const ***destTableName*** String) Logical

Compatibility Procedure
rename (const ***tableName*** String, const ***destTableName*** String) Logical

replaceItem

Array *Method*
Overwrites an item in an array with another item.
replaceItem (const ***keyItem*** AnyType, const ***newItem*** AnyType)

reSync

UIObject *Method*
Resynchronizes an object to a TCursor.
reSync (const ***tc*** TCursor) Logical

retryPeriod

Session *Method/Procedure*
Returns the number of seconds to retry an operation on a locked record or table.
retryPeriod ( ) SmallInt

rgb

UIObject *Procedure*
Defines a color.
rgb (const ***red*** SmallInt, const ***green*** SmallInt, const ***blue*** SmallInt) LongInt

round

Number *Method*
Rounds a number to a specified number of decimal places.
round (const ***places*** SmallInt) Number

rTrim

String *Method*
Removes trailing blanks from a string.
rTrim ( ) String

run

Form *Method*
Runs a form that is currently open in the Form Design window.
run ( ) Logical

Report *Method*
Runs a report that is currently open in the Report Design window.
run ( ) Logical

save

Form *Method*

Report *Method*
Saves a form or report to disk.
save ([const ***newFormName*** String]) Logical

saveCFG

Session *Method/Procedure*
Saves the current session's alias information to a file.
saveCfg (const ***fileName*** String) Logical

search

String *Method*
Returns the position of one string inside another.
search (const ***str*** String) SmallInt

second

DateTime *Method*

Time *Method*

Extracts as a number the seconds from a Time or DateTime.

second ( ) SmallInt

seqNo

TCursor *Method*

Returns the sequential record number of the current record.

seqNo ( ) LongInt

setAliasPath

Session *Method/Procedure*

Sets the path for an alias.

setAliasPath (const ***aliasName*** String, const ***aliasPath*** String) Logical

setAltKeyDown

KeyEvent *Method*

Simulates pressing and holding *Alt* during a KeyEvent.

setAltKeyDown (const ***yesNo*** Logical)

setChar

KeyEvent *Method*

Specifies an ANSI character for a KeyEvent.

setChar (const ***char*** String)

setControlKeyDown

KeyEvent *Method*

Simulates pressing and holding *Ctrl* during a KeyEvent.

setControlKeyDown (const ***yesNo*** Logical)

MouseEvent *Method*

Simulates pressing and holding *Ctrl* during a MouseEvent.

setControlKeyDown (const ***yesNo*** Logical)

setData

MenuEvent *Method*

Specifies information about a MenuEvent.

setData (const ***menuData*** LongInt)

setDir

FileSystem *Method*

Sets the directory path for a FileSystem variable.

setDir (const ***name*** String) Logical

setDrive

FileSystem *Method*

Makes a specified drive the default drive.

setDrive (const ***name*** String) Logical

setErrorCode

ActionEvent *Method*

ErrorEvent *Method*

Event *Method*

KeyEvent *Method*

MenuEvent *Method*

MouseEvent *Method*

MoveEvent *Method*

StatusEvent *Method*

TimerEvent *Method*

ValueEvent *Method*

Sets the error code for an Event.

setErrorCode (const ***errorId*** LongInt)

setExclusive

Table *Method*

Specifies whether to give the user exclusive rights to a table when it is opened.

setExclusive ([const ***yesNo*** Logical])

setFieldValue

TCursor *Method*

Assigns a value to a specified field.

1. **setFieldValue** (const ***fieldName*** String, const ***value*** AnyType) Logical
2. **setFieldValue** (const ***fieldNum*** SmallInt, const ***value*** AnyType) Logical

setFileAccessRights

FileSystem *Procedure*

Sets access rights (also called attributes) of a file.

setFileAccessRights (const ***fileName*** String , const ***rights*** String) Logical

setFilter

Table *Method*
Sets the range of records a table can point to.
setFilter ([const ***exactMatchVal*** AnyType,] *
const ***minVal*** AnyType, const ***maxVal*** AnyType) Logical

TCursor *Method*
Sets the range of records a TCursor can point to.
setFilter ([const ***exactMatchVal*** AnyType,] *
const ***minVal*** AnyType, const ***maxVal*** AnyType) Logical

UIObject *Method*
Sets the range of records a table object can point to.
setFilter ([const ***exactMatchVal*** AnyType,] *
const ***minVal*** AnyType, const ***maxVal*** AnyType) Logical

setFlyAwayControl

TCursor *Method*
Controls whether the TCursor points to a record whose position has changed as the result of an **unlockRecord**.
setFlyAwayControl ([const ***yesNo*** Logical]) Logical

setId

ActionEvent *Method*
Specifies the ID of an ActionEvent.
setId (const ***actionId*** SmallInt)

MenuEvent *Method*
Specifies the ID of a MenuEvent.
setId (const ***actionId*** SmallInt)

setIndex

Table *Method*
Specifies an index for a table.
1. (Paradox tables) **setIndex** (const ***indexName*** String) Logical
2. (dBASE tables) **setIndex** (const ***indexName*** String
[, const ***tagName*** String]) Logical

setInside

MouseEvent *Method*
Sets the mouse to be inside the current object.
setInside (const ***trueFalse*** Logical) Logical

setItem

DDE *Method*

Specifies an item in a DDE conversation.

setItem (const ***server*** String)

setLeftDown

MouseEvent *Method*

Simulates pressing the left mouse button.

setLeftDown (const ***yesNo*** Logical)

setMenuChoiceAttribute

Menu *Procedure*

Sets the display attribute of a menu item.

setMenuChoiceAttribute (const ***menuChoice*** String,
const ***menuAttribute*** SmallInt)

setMenuChoiceAttributeById

Menu *Procedure*

Sets the display attribute of a menu item.

setMenuChoiceAttributeById (const ***menuId*** SmallInt,
const ***menuAttribute*** SmallInt)

setMiddleDown

MouseEvent *Method*

Simulates pressing the middle mouse button.

setMiddleDown (const ***yesNo*** Logical)

setMousePosition

MouseEvent *Method*

Sets the position of the pointer for an event.

1. **setMousePosition** (const ***xPosition*** LongInt, const ***yPosition*** LongInt)
2. **setMousePosition** (const ***p*** Point)

setMouseScreenPosition

System *Procedure*

Displays the pointer at a specified position.

1. **setMouseScreenPosition** (const ***mousePosition*** Point)
2. **setMouseScreenPosition** (const ***x*** LongInt, const ***y*** LongInt)

setMouseShape

System *Procedure*

Specifies the shape of the pointer.

setMouseShape (const ***mouseShapeId*** LongInt) LongInt

setNewValue

ValueEvent *Method*
Specifies a value to set for a ValueEvent.
setNewValue (const ***newValue*** AnyType)

setPosition

Application *Method*

Form *Method/Procedure*

Report *Method*

TableView *Method*
Positions a window onscreen.
setPosition (const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt, const ***h*** LongInt)

TextStream *Method*
Positions the pointer in a text file.
setPosition (const ***offset*** LongInt)

UIObject *Method*
Sets the position of an object.
setPosition (const ***x*** LongInt, const ***y*** LongInt, const ***w*** LongInt, const ***h*** LongInt)

setProperty

UIObject *Method*
Sets a property to a specified value.
setProperty (const ***propertyName*** String, const ***propertyValue*** AnyType)

setReadOnly

Table *Method*
Specifies whether to give the user read-only rights to a table when it is opened.
setReadOnly ([const ***yesNo*** Logical])

setReason

ActionEvent *Method*

Event *Method*

KeyEvent *Method*

MouseEvent *Method*

TimerEvent *Method*

ValueEvent *Method*
Specifies a Reason for an event.
setReason (const ***reasonId*** SmallInt)

ErrorEvent *Method*
Specifies a Reason for an ErrorEvent.
setReason (const ***reasonId*** SmallInt)

MenuEvent *Method*
Specifies a Reason for a MenuEvent.
setReason (const ***reasonId*** SmallInt)

MoveEvent *Method*
Specifies a Reason for a MoveEvent.
setReason (const ***reasonId*** SmallInt)

StatusEvent *Method*
Specifies a Reason for a StatusEvent.
setReason (const ***reasonId*** SmallInt)

setRetryPeriod

Session *Method/Procedure*
Sets the number of seconds to retry an action on a locked table or record.
setRetryPeriod (const ***period*** SmallInt) Logical

setRightDown

MouseEvent *Method*
Simulates pressing the right mouse button.
setRightDown (const ***yesNo*** Logical)

setShiftKeyDown

KeyEvent *Method*
Simulates pressing and holding *Shift* during a KeyEvent.
setShiftKeyDown (const ***yesNo*** Logical)

MouseEvent *Method*
Simulates pressing and holding *Shift* during a MouseEvent.
setShiftKeyDown (const ***yesNo*** Logical)

setSize

Array *Method*
Specifies the size of an array.
setSize (const ***size*** LongInt)

setStatusValue

StatusEvent *Method*
Specifies the text of a status message.
setStatusValue (const ***statusValue*** AnyType)

setTimer

UIObject *Method*
Starts the timer for an object.
setTimer (const ***milliSeconds*** LongInt [, const ***repeat*** Logical])

setTitle

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*
Sets the text in the window title bar.
setTitle (const ***text*** String)

setVChar

KeyEvent *Method*
Specifies a Windows virtual character for a KeyEvent.
setVChar (const ***char*** String)

setVCharCode

KeyEvent *Method*
Specifies a Windows virtual character for a KeyEvent.
setVCharCode (const ***vk_Constant*** SmallInt)

setX

MouseEvent *Method*
Specifies the horizontal coordinate of the pointer position.
setX (const ***xPosition*** LongInt)

Point *Method*
Specifies the horizontal coordinate of a point.
setX (const ***newXValue*** LongInt)

setXY

Point *Method*
Specifies the horizontal and vertical coordinates of a point.
setXY (const ***newXValue*** LongInt, const ***newYValue*** LongInt)

setY

MouseEvent *Method*
Specifies the vertical coordinate of the pointer position.
setY (const ***yPosition*** LongInt)

Point *Method*
Specifies the vertical coordinate of a point.
setY (const ***newYValue*** LongInt)

show

Application *Method*

Form *Method/Procedure*

Report *Method*

TableView *Method*
Displays a minimized window at its previous size; makes a hidden window visible.
show ( )

Menu *Method*
Displays a menu.
show ( )

PopUpMenu *Method*
Displays a pop-up menu and returns the item selected.
show ([const ***xTwips*** SmallInt, const ***yTwips*** SmallInt]) String

showDeleted

Table *Method*
Specifies whether to display deleted records in a dBASE table.
showDeleted ([const ***yesNo*** Logical]) Logical

TCursor *Method*
Specifies whether to show deleted records in a dBASE table.
showDeleted ([const ***yesNo*** Logical]) Logical

showSpeedBar

Form *Procedure*
Makes the SpeedBar visible.
showSpeedBar ( )

sin

Number *Method*
Returns the sine of an angle.
sin ( ) Number

sinh

Number *Method*
Returns the hyperbolic sine of an angle.
sinh ( ) Number

size

Array *Method*
Returns the number of items in an Array.
size ( ) LongInt

Binary *Method*
Returns the number of bytes in a Binary variable.
size ( ) LongInt

DynArray *Method*
Returns the number of elements in a DynArray.
size ( ) LongInt

FileSystem *Method*
Returns the size of a file.
size ( ) LongInt

String *Method*
Returns the length of a string.
size ( ) SmallInt

TextStream *Method*
Returns the number of characters in a text file.
size ( ) LongInt

skip

TCursor *Method*
Moves forward or backward a specified number of records in a table.
skip ([const ***nRecords*** LongInt]) Logical

UIObject *Method*
Moves forward or backward a specified number of records in a table.
skip (const ***nRecords*** LongInt) Logical

sleep

System *Procedure*
Produces a delay of a specified duration.
sleep ([const ***numberOfMilliSeconds*** LongInt])

smallInt

SmallInt *Procedure*
Casts a value as a small integer.
smallInt (const ***value*** AnyType) SmallInt

sort

Table *Keyword*

Sorts a table.

sort ***sourceTable***
  [**on** *fieldNameList* [**D**]]
  [**to** *destTable*]
endSort

sortTo

TCursor *Method*

Sorts a table.

1. **sortTo** (const ***destTableName*** String, const ***numFields*** SmallInt,
   const ***sortFields*** Array[] String,
   const ***sortOrder*** Array[] SmallInt) Logical

2. **sortTo** (const ***destTable*** Table, const ***numFields*** SmallInt,
   const ***sortFields*** Array[] String,
   const ***sortOrder*** Array[] SmallInt) Logical

sound

System *Procedure*

Creates a sound of specified frequency and duration.

sound (const ***freqHertz*** LongInt, const ***durationMilliSecs*** LongInt)

space

String *Procedure*

Creates a string of a specifed number of spaces.

space (const ***numberOfSpaces*** SmallInt) String

splitFullFileName

FileSystem *Procedure*

Breaks a full path name into its component parts.

1. **splitFullFileName** (const ***fullFileName*** String,
   var ***components*** DynArray[] String)

2. **splitFullFileName** (const ***fullFileName*** String,
   var ***driveName*** String, var ***pathName*** String,
   var ***fileName*** String, var ***extensionName*** String)

sqrt

Number *Method*

Returns the square root of a number.

sqrt ( ) Number

startUpDir

FileSystem *Procedure*

Returns a string containing the path to the user's start-up directory.

startUpDir ( ) String

statusValue

StatusEvent *Method*

Returns the text of a status message.

statusValue ( ) AnyType

string

String *Procedure*

Casts a value as a String.

string (const ***value*** AnyType [, const ***value*** AnyType]*) String

strVal

String *Procedure*

Converts a value to a string.

strVal (const ***value*** AnyType) String

subStr

String *Method*

Returns a portion of a string.

subStr (const ***startIndex*** Number [, const ***numberOfChars*** SmallInt]) String

subtract

Table *Method*

Subtracts the records in one table from another table.

1. **subtract** (const ***destTableName*** String) Logical
2. **subtract** (const ***destTable*** Table) Logical

Compatibility Procedure

subtract (const ***tableName*** String, const ***destTableName*** String) Logical

TCursor *Method*

Subtracts the records in one table from another table.

1. **subtract** (const ***destTableName*** String) Logical
2. **subtract** (const ***destTable*** Table) Logical
3. **subtract** (const ***destTable*** TCursor) Logical

switchIndex

TCursor *Method*

Specifies another index to use to view the records in a table.

1. **switchIndex** (const ***indexName*** String
   [, const ***stayOnRecord*** Logical]) Logical

2. **switchIndex** (const ***indexFileName*** String
   [, const ***tagName*** String [, const ***stayOnRecord*** Logical]]) Logical

UIObject *Method*

Specifies another index to use to view the records in a table.

1. **switchIndex** (const ***indexName*** String
   [, const ***stayOnRecord*** Logical]) Logical

2. **switchIndex** (const ***indexFileName*** String
   [, const ***tagName*** String [, const ***stayOnRecord*** Logical]]) Logical

switchMenu

PopUpMenu *Keyword*

Builds and displays a pop-up menu and handles the menu choice.

switchMenu
  caseList
  [**otherwise : *Statements***]
endSwitchMenu

caseList is any number of statements in the following form:
CASE *menuItem* : *Statements*

sysInfo

System *Procedure*

Creates a dynamic array of information about the system running Paradox.

sysInfo (var ***info*** DynArray[] AnyType)

tableName

TCursor *Method*

Returns the name of the table associated with a TCursor.

tableName ( ) String

tableRights

Table *Method*
Specifies whether the user has rights to perform certain operations on a table.
tableRights (const ***rights*** String) Logical

Compatibility Procedure
tableRights (const ***tableName*** String, const ***rights*** String) Logical

TCursor *Method*
Reports about the operations you can perform on a table.
tableRights (const ***rights*** String) Logical

tan

Number *Method*
Returns the tangent of an angle.
tan ( ) Number

tanh

Number *Method*
Returns the hyperbolic tangent of an angle.
tanh ( ) Number

time

FileSystem *Method*
Returns the time and date a file was last modified.
time ( ) DateTime

Time *Procedure*
Casts a value as a time, or returns the current time.
time ([const ***value*** AnyType]) Time

toANSI

String *Method*
Converts a string of OEM characters to ANSI characters.
toANSI ( ) String

today

Date *Procedure*
Returns the current date.
today ( ) Date

toOEM

String *Method*
Converts a string of ANSI characters to OEM characters.
toOEM ( ) String

totalDiskSpace

FileSystem *Method*
Returns the capacity of a drive.
totalDiskSpace (const ***driveLetter*** String) LongInt

tracerClear

System *Procedure*
Clears the Tracer window.
tracerClear ( )

tracerHide

System *Procedure*
Hides the Tracer window.
tracerHide ( )

tracerOff

System *Procedure*
Closes the Tracer window.
tracerOff ( )

tracerOn

System *Procedure*
Activates code tracing.
tracerOn ( )

tracerSave

System *Procedure*
Saves the contents of the Tracer window to a file.
tracerSave (const ***fileName*** String)

tracerShow

System *Procedure*
Makes the Tracer window visible.
tracerShow ( )

tracerToTop

System *Procedure*
Makes the Tracer window the topmost window.
tracerToTop ( )

tracerWrite

System *Procedure*
Writes a message to the Tracer window.
tracerWrite (const ***message*** String [, const ***message*** String]*)

truncate

Number *Method*
Truncates a number to a specified number of decimal places.
truncate (const ***places*** SmallInt) Number

twipsToPixels

System *Procedure*
Converts screen coordinates from twips to pixels.
twipsToPixels (const ***twips*** Point) Point

UIObject *Method*
Converts screen coordinates from twips to pixels.
twipsToPixels (const ***twips*** Point) Point

type

Table *Method*
Returns the type of a table.
type ( ) String

TCursor *Method*
Returns the type of a table.
type ( ) String

unAssign

AnyType *Method*
Sets a variable's state to unassigned.
unAssign ( ) Logical

unAttach

Table *Method*
Ends the association between a Table variable and a table on disk.
unAttach ( ) Logical

unDeleteRecord

TCursor *Method*
Undeletes the current record from a dBASE table.
unDeleteRecord ( ) Logical

UIObject *Method*
Undeletes the current record from a dBASE table.
unDeleteRecord ( ) Logical

unlock

Session *Procedure*
Unlocks one or more tables.
unlock (const ***table*** { Table | TCursor | String}, const ***lockType*** String [, const ***table*** { Table | TCursor | String }, const ***lockType*** String]*) Logical

Table *Method*
Unlocks a specified table.
unlock (const ***lockType*** String) Logical

TCursor *Method*
Removes specified locks from a TCursor.
unlock (const ***lockType*** String) Logical

unlockRecord

TCursor *Method*
Unlocks the current record.
unlockRecord ( ) Logical

UIObject *Method*
Removes a write lock from the current record.
unlockRecord ( ) Logical

unProtect

Table *Method*
Decrypts and removes an owner password from a table.
unProtect ([const ***password*** String]) Logical

Compatibility Procedure
unProtect (const ***tableName*** String [, const ***password*** String]) Logical

updateRecord

TCursor *Method*
Updates the existing record with data from the new record when a key violation exists.
updateRecord ([const ***moveTo*** Logical]) Logical

upper

String *Method*
Converts a string to uppercase.
upper ( ) String

usesIndexes

Table *Method*
Specifies index files to use and maintain with a dBASE table.
usesIndexes (const ***indexFileName*** String
[, const ***indexFileName*** String]* Logical

vChar

KeyEvent *Method*
Returns a Windows virtual character.
vChar ( ) String

vCharCode

KeyEvent *Method*
Returns the integer value of a Windows virtual character.
vCharCode ( ) SmallInt

version

System *Procedure*
Returns the Paradox version number.
version ( ) String

view

AnyType *Method*
Currency *Method*
Date *Method*
DateTime *Method*
Logical *Method*
LongInt *Method*
Number *Method*
Point *Method*
SmallInt *Method*
String *Method*
Time *Method*
Displays in a dialog box the value of a variable or object.
view ([const ***title*** String])

Array *Method*
Displays in a dialog box the contents of an Array; the displayed Array cannot be modified.
view ([const ***title*** String])

DynArray *Method*
Displays the contents of a DynArray in a dialog box; the displayed DynArray cannot be modified.
view ([const ***title*** String])

Record *Method*
Displays in a dialog box the value of a Record; the displayed Record cannot be modified.
view ([const ***title*** String])

UIObject *Method*
Displays the value of an object in a dialog box.
view ([const ***title*** String])

---

vkCodeToKeyName

String *Procedure*
Converts a virtual key-code constant to a virtual key-code string.
vkCodeToKeyName (const ***vkCode*** SmallInt) String

---

wait

Form *Method*
Suspends execution of a method.
wait ( ) AnyType

TableView *Method*
Suspends execution of a method.
wait ( )

---

wasLastClicked

UIObject *Method*
Tells if an object was the last object to receive a mouse click.
wasLastClicked ( ) Logical

---

wasLastRightClicked

UIObject *Method*
Tells if an object was the last object to receive a right mouse click.
wasLastRightClicked ( ) Logical

---

windowClientHandle

Application *Method*

Form *Method/Procedure*
Returns the handle of a window.
windowClientHandle ( ) SmallInt

windowHandle

Application *Method*
Form *Method/Procedure*
Report *Method*
TableView *Method*
Returns the handle of a window.
windowHandle ( ) SmallInt

windowsDir

FileSystem *Procedure*
Returns the path to the Windows directory.
windowsDir ( ) String

windowsSystemDir

FileSystem *Procedure*
Returns the path to the SYSTEM directory for Windows.
windowsSystemDir ( ) String

winGetMessageID

System *Procedure*
Returns the ID of a Windows message.
winGetMessageID (const ***msgName*** SmallInt) SmallInt

winPostMessage

System *Procedure*
Posts a message to Windows.
winPostMessage (const ***hWnd*** SmallInt, const ***msg*** SmallInt, const ***wParam*** SmallInt, const ***lParam*** LongInt) Logical

winSendMessage

System *Procedure*
Sends a message to Windows.
winSendMessage (const ***hWnd*** SmallInt, const ***msg*** SmallInt, const ***wParam*** SmallInt, const ***lParam*** LongInt) Logical

workingDir

FileSystem *Procedure*
Returns the name of the current working directory.
workingDir ( ) String

writeEnvironmentString

System *Procedure*
Writes information about the user's system environment to a file.
writeEnvironmentString (const ***key*** String, const ***value*** String) Logical

writeLine

TextStream *Method*
Writes a string to a text file.
writeLine (const ***value*** AnyType [, const ***value*** AnyType]*) Logical

writeProfileString

System *Procedure*
Writes information about the user's system to a file.
writeProfileString (const ***fileName*** String, const ***section*** String, const ***key*** String, const ***value*** String) Logical

writeQBE

Database *Method/Procedure*
Writes a query statement or a query string to a file.
1. **writeQBE** (const ***qbeVar*** Query, const ***fileName*** String) Logical
2. **writeQBE** (const ***str*** String, const ***fileName*** String) Logical

Query *Method*
Writes a query statement to a specified file.
writeQBE (const ***fileName*** String) Logical

writeString

TextStream *Method*
Writes a character string to a text file.
writeString (const ***value*** AnyType [, const ***value*** AnyType]*) Logical

writeToClipboard

Graphic *Method*
Writes a bitmap to the Clipboard.
writeToClipboard ( ) Logical

OLE *Method*
Copies an OLE variable to the Clipboard.
writeToClipboard ( ) Logical

writeToFile

Binary *Method*
Writes the data stored in a Binary variable to a disk file.
writeToFile (const ***fileName*** String) Logical

Graphic *Method*
Writes a bitmap to a file.
writeToFile (const ***fileName*** String) Logical

Memo *Method*
Writes a memo to a file.
writeToFile (const ***fileName*** String) Logical

x

MouseEvent *Method*
Returns the horizontal coordinate of the pointer position.
x ( ) LongInt

Point *Method*
Returns the horizontal coordinate of a point.
x ( ) LongInt

y

MouseEvent *Method*
Returns the vertical coordinate of the pointer position.
y ( ) LongInt

Point *Method*
Returns the vertical coordinate of a point.
y ( ) LongInt

year

Date *Method*

DateTime *Method*
Extracts as a number the year from a DateTime.
year ( ) SmallInt

PARADOX®
FOR WINDOWS

BORLAND

Corporate Headquarters: 1800 Green Hills Road. P.O. Box 660001, Scotts Valley, CA 95067-0001, (408) 438-8400. Offices in: Australia, Belgium, Canada, Denmark, France, Germany, Hong Kong, Italy, Japan, Korea, Malaysia, Netherlands, New Zealand, Singapore, Spain, Sweden, Taiwan, and United Kingdom ▪ Part # PDX1110WW21796 ▪ BOR 2714