Powersoft

)pen Tools from Sybase, Inc.

)owerBuilder :oundation Class Library

)bject Reference: Volume 2

/ersion 6

PowerBuilder®

AA0819

October 1997

Copyright © 1991-1997 Sybase, Inc. and its subsidiaries.
All rights reserved.
Printed in the United States of America.

Information in this manual may change without notice and does not represent a commitment on the part of Sybase, Inc. and its subsidiaries.

The software described in this manual is provided by Powersoft Corporation under a Powersoft License agreement. The software may be used only in accordance with the terms of the agreement.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc. and its subsidiaries.

Sybase, Inc. and its subsidiaries claim copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of other rights of Sybase, Inc. and its subsidiaries.

ClearConnect, Column Design, ComponentPack, InfoMaker, ObjectCycle, PowerBuilder, PowerDesigner, Powersoft, S-Designor, SQL SMART, and Sybase are registered trademarks of Sybase, Inc. and its subsidiaries. Adaptive Component Architecture, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Warehouse, AppModeler, DataArchitect, DataExpress, Data Pipeline, DataWindow, dbQueue, ImpactNow, InstaHelp, Jaguar CTS, jConnect for JDBC, MetaWorks, NetImpact, Optima++, Power++, PowerAMC, PowerBuilder Foundation Class Library, Power J, PowerScript, PowerSite, Powersoft Portfolio, Powersoft Professional, PowerTips, ProcessAnalyst, Runtime Kit for Unicode, SQL Anywhere, The Model For Client/Server Solutions, The Future Is Wide Open, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, Viewer, WarehouseArchitect, Watcom, Watcom SQL Server, Web.PB, and Web.SQL are trademarks of Sybase, Inc. or its subsidiaries. Certified PowerBuilder Developer and CPD are service marks of Sybase, Inc. or its subsidiaries. DataWindow is a patented proprietary technology of Sybase, Inc. or its subsidiaries.

AccuFonts is a trademark of AccuWare Business Solutions Ltd.

All other trademarks are the property of their respective owners.

Contents

About This Book ix

VOLUME 1

1 **Object Reference Conventions** 1
- Naming conventions 2
- Online Help 9
- Sample application and code examples 10

2 **Window Objects** 11
- w_child 12
- w_frame 13
- w_main 20
- w_master 21
- w_popup 58
- w_response 59
- w_sheet 62

3 **Menus** 65
- m_dw 66
- m_edit 69
- m_frame 71
- m_lvs 73
- m_master 76
- m_oc 82
- m_tvs 84

4 **Global Functions** 87
- f_SetFilesrv 88
- f_SetPlatform 89

5 **Global Structures and Structure Objects 91**
n_cst_aboutattrib 92
n_cst_baseattrib 93
n_cst_calculatorattrib 94
n_cst_calendarattrib 95
n_cst_columnattrib 96
n_cst_dberrorattrib 97
n_cst_dirattrib 98
n_cst_dwobjectattrib 99
n_cst_dwpropertyattrib 100
n_cst_errorattrib 101
n_cst_filterattrib 102
n_cst_findattrib 103
n_cst_infoattrib 104
n_cst_itemattrib 105
n_cst_linkageattrib 106
n_cst_logonattrib 107
n_cst_lvsrvattrib 108
n_cst_propertyattrib 109
n_cst_restorerowattrib 110
n_cst_returnattrib 111
n_cst_selectionattrib 112
n_cst_sortattrib 113
n_cst_splashattrib 114
n_cst_sqlattrib 115
n_cst_textstyleattrib 116
n_cst_tmgregisterattrib 117
n_cst_toolbarattrib 118
n_cst_tvsrvattrib 119
n_cst_zoomattrib 120
s_pagesetupattrib 121
s_paperattrib 122
s_printdlgattrib 123
s_svalue 124

6 **Standard Visual User Objects 125**
u_cb 126
u_cbx 129
u_ddlb 132
u_ddplb 139
u_dw 146
u_em 198
u_gb 206
u_gr 207

u_hsb 210
u_lb 213
u_lvs 218
u_mle 251
u_oc 257
u_p 265
u_pb 268
u_plb 271
u_rb 276
u_rte 279
u_sle 306
u_st 312
u_tab 315
u_tvs 331
u_vsb 369

7 **Custom Visual User Objects 373**
u_base 374
u_calculator 391
u_calendar 416
u_progressbar 457
u_st_splitbar 485
u_tabpg 501

8 **Standard Class User Objects 503**
n_cn 504
n_cxinfo 508
n_cxk 509
n_dda 510
n_ds 511
n_dsa 535
n_err 536
n_inet 538
n_ir 539
n_ms 540
n_msg 542
n_oo 544
n_ostg 545
n_ostm 546
n_pl 547
n_srv 549
n_tmg 550
n_tr 553

n_trp 569

VOLUME 2

9 **Custom Class User Objects 571**
n_base 572
n_cst_appmanager 575
n_cst_apppreference 607
n_cst_color 627
n_cst_conversion 629
n_cst_datetime 642
n_cst_debug 661
n_cst_dropdown 669
n_cst_dssrv 675
n_cst_dssrv_multitable 676
n_cst_dssrv_printpreview 677
n_cst_dssrv_report 678
n_cst_dwcache 679
n_cst_dwsrv 695
n_cst_dwsrv_dropdownsearch 723
n_cst_dwsrv_filter 735
n_cst_dwsrv_find 747
n_cst_dwsrv_linkage 764
n_cst_dwsrv_multitable 832
n_cst_dwsrv_printpreview 844
n_cst_dwsrv_property 854
n_cst_dwsrv_querymode 858
n_cst_dwsrv_report 870
n_cst_dwsrv_reqcolumn 917
n_cst_dwsrv_resize 925
n_cst_dwsrv_rowmanager 943
n_cst_dwsrv_rowselection 954
n_cst_dwsrv_sort 972
n_cst_error 989
n_cst_filesrv 1019
n_cst_filesrvmac 1050
n_cst_filesrvsol2 1071
n_cst_filesrvwin16 1093
n_cst_filesrvwin32 1110
n_cst_inifile 1133
n_cst_luw 1139
n_cst_lvsrv 1157
n_cst_lvsrv_datasource 1164

n_cst_lvsrv_sort 1203
n_cst_menu 1210
n_cst_metaclass 1216
n_cst_numerical 1223
n_cst_platform 1230
n_cst_platformmac 1239
n_cst_platformsol2 1246
n_cst_platformwin16 1256
n_cst_platformwin32 1266
n_cst_resize 1276
n_cst_rtefind 1287
n_cst_security 1294
n_cst_selection 1298
n_cst_sql 1303
n_cst_sqlspy 1306
n_cst_string 1312
n_cst_trregistration 1334
n_cst_tvsrv 1341
n_cst_tvsrv_levelsource 1351
n_cst_tvsrv_print 1401
n_cst_winsrv 1405
n_cst_winsrv_preference 1408
n_cst_winsrv_sheetmanager 1427
n_cst_winsrv_statusbar 1435

About This Book

Subject

This book describes the objects in the PowerBuilder Foundation Class Library (PFC). Each object's discussion includes:

- A diagram providing a quick overview of the object's capabilities
- Instance variables
- Events and user events
- Object functions

Online Help

The PFC online Help (pbpfc60.hlp) contains all the objects, events, and functions explained in this book. This Help file includes customized buttons and links to help you navigate through inheritance hierarchies and related objects.

FOR INFO For PFC usage information, see the *PowerBuilder Foundation Class Library User's Guide*.

Audience

This book is for experienced PowerBuilder developers. It assumes that:

- You are currently developing applications using PowerBuilder and understand the concepts and techniques described in the PowerBuilder *Application Techniques* book
- You understand SQL and how to use your site-specific DBMS

CHAPTER 9

Custom Class User Objects

About this chapter

This chapter describes the custom class user objects in PFC.

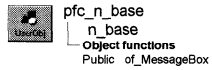
Contents

The user objects are listed in alphabetical order. Each object's discussion includes alphabetical listings of instance variables, events, and object functions.

n_base

Description — Ancestor object for all PFC custom class user objects.

Ancestry



Library — PFCMAIN.PBL
PFEMAIN.PBL

Usage — PFC uses this object as the ancestor for all custom class user objects.

Descendants — All PFC custom class user objects descend from n_base.

See also — u_base

Instance variables

N_base includes instance variables:

Instance variable	Description	Data type	Access	Usage
CONTINUE_ACTION	Constant set to 1	Integer	Public	Internal
FAILURE	Constant set to -1	Integer	Public	Internal
NO_ACTION	Constant set to 0	Integer	Public	Internal
PREVENT_ACTION	Constant set to 0	Integer	Public	Internal
SUCCESS	Constant set to 1	Integer	Public	Internal

Functions

N_base includes a precoded function:

of_MessageBox

of_MessageBox

Description — Displays a MessageBox.

Access

Protected

Syntax

instancename.**of_MessageBox** (*id, title, message, icon, button, default*)

Argument	Description
instancename	Instance name of n_base
id	String specifying an ID for the message
title	String specifying the title of the MessageBox
message	String specifying the message text
icon	Icon enumerated data type indicating the icon you want to display on the left side of the message box. Values are: ♦ Information! ♦ StopSign! ♦ Exclamation! ♦ Question! ♦ None!
button	Button enumerated data type indicating the set of CommandButtons you want to display at the bottom of the message box. The buttons are numbered in the order listed in the enumerated data type. Values are: ♦ OK! ♦ OKCancel! ♦ YesNo! ♦ YesNoCancel! ♦ RetryCancel! ♦ AbortRetryIgnore!
default	Integer specifying the number of the button you want to be the default button

Return value

Integer. Returns 1 if the function succeeds -1 if an error occurs.

Usage

Override this function to control MessageBox behavior in PFC custom class user objects. You can either override this function in n_base (which affects all PFC custom class user objects) or override it in the extension level of an individual custom class user object.

The *id* argument is not used in the default implementation.

Examples

This example is from the n_cst_luw of_DBError function:

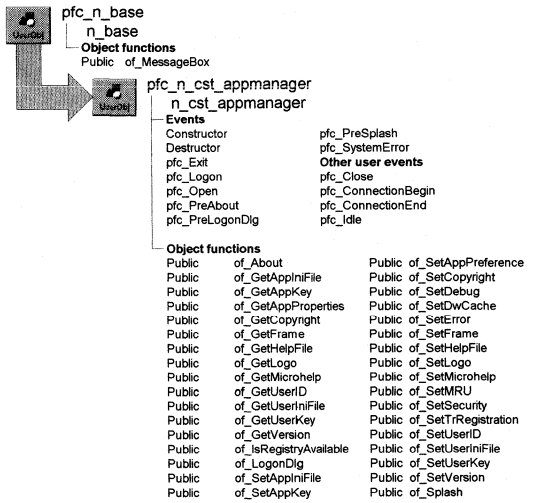
```
of_Messagebox('pfc_luw_dberror','Save', &
   as_error, StopSign!, Ok!, 1)
...
```

n_cst_appmanager

Description

N_cst_appmanager is the application manager and functions as a substitute for the Application object. It contains many useful functions as well as reference variables for application services.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_apppreference
n_cst_debug
n_cst_dwcache
n_cst_error
n_cst_mru
n_cst_security
n_cst_trregistration
w_about
w_frame
w_logon
w_splash

Usage

To use n_cst_appmanager:

1 (Optional) Access the extension-level n_cst_appmanager user object and create a descendant application manager.

2 Define a global variable, gnv_app, of type n_cst_appmanager (or your customized descendant).

```
n_cst_appmanager   gnv_app
```

3 Add PowerScript code to your application's Open event to create an instance of n_cst_appmanager and call the pfc_Open event, passing the commandline argument:

```
gnv_app = CREATE n_cst_appmanager
gnv_app.Event pfc_Open(commandline)
```

4 Add PowerScript code to the Constructor event to selectively enable application services:

Preference service—of_SetAppPreference
Debug service—of_SetDebug
DataWindow caching service—of_SetDwCache
Most recently used object service—of_SetMRU
Error handling service—of_SetError
Transaction registration service—of_SetTrRegistration
Security service—of_SetSecurity

5 Add PowerScript code to the pfc_Open event to open the initial window.

6 Add code to the Constructor event in n_cst_appmanager (or your customized descendant) to initialize instance variables for version, company, and INI file.

See also

n_cst_apppreference
n_cst_debug
n_cst_dwcache
n_cst_error
n_cst_mru
n_cst_security
n_cst_trregistration

Instance variables

N_cst_appmanager includes instance variables:

Instance variable	Description	Data type	Access	Usage
iapp_object	Application object associated with n_cst_appmanager	Application	Public	Use to access application attributes

Instance variable	Description	Data type	Access	Usage
ib_microhelp	Reports whether the application supports MicroHelp	Boolean	Protected	Call of_GetMicrohelp and of_SetMicrohelp to access
ienv_object	The Environment object, which contains information such as screen height, screen width, and operating system	Environment	Public	Use instead of repeated calls to the GetEnvironment function
inv_appprefe	Reference variable for the preference service	n_cst_apppreference	Public	Use in dot notation to access n_cst_apppreference functions
inv_debug	Reference variable for the debug service	n_cst_debug	Public	Use in dot notation to access n_cst_debug functions
inv_dwcache	Reference variable for data caching services	n_cst_dwcache	Public	Use in dot notation to access n_cst_dwcache functions
inv_error	Reference variable for the error service	n_cst_error	Public	Use in dot notation to access n_cst_error functions
inv_mru	Reference variable for the most recently used object service	n_cst_mru	Public	Use in dot notation to access n_cst_mru functions
inv_security	Reference variable for the security service	n_cst_security	Public	Use in dot notation to access n_cst_security functions
inv_trregistration	Reference variable for the transaction registration service	n_cst_trregistration	Public	Use in dot notation to access n_cst_trregistration functions

Instance variable	Description	Data type	Access	Usage
is_appinifile	The name of the application INI file	String	Protected	Use in profile functions (such as ProfileString) to specify the name of the application INI file (typically includes database connection profile) Call of_GetAppIniFile and of_SetAppIniFile to access
is_appkey	The name of the registry key for application information	String	Protected	Use in PowerScript functions (such as RegistryGet) to specify the name of the application's registry key (typically includes database connection profile) Call of_GetAppKey and of_SetAppKey to access
is_copyright	Application-specific copyright information	String	Protected	Call of_GetCopyright and of_SetCopyright to access The w_about window displays this variable automatically
is_helpfile	The name of the application's Help file	String	Protected	Use in the ShowHelp function Call of_GetHelpFile and of_SetHelpFile access
is_logo	The name of the bitmap displayed in the application's About box	String	Protected	Call of_GetLogo and of_SetLogo to access
is_userid	User information	String	Protected	Call of_GetUserID and of_SetUserID to access

Instance variable	Description	Data type	Access	Usage
is_userinifile	The name of the user INI file	String	Protected	Use in profile functions (such as ProfileString) to access information in the user INI file (typically contains user-specific information, such as toolbar settings and window position) Call of_GetUserIniFile and of_SetUserIniFile to access
is_userkey	The name of the registry key for user information	String	Protected	User information (typically includes toolbar settings and window position) Call of_GetUserKey and of_SetUserKey to access
is_version	Application-specific version information	String	Protected	Call of_GetVersion and of_SetVersion to access The w_about window displays this variable automatically
iw_frame	Frame window	w_frame	Protected	Use to get the active frame window This is set automatically in the w_frame Activate event Call of_GetFrame to access

Events

N_cst_appmanager includes precoded events:

Constructor	pfc_Logon
Destructor	pfc_Open
pfc_Close	pfc_PreAbout
pfc_ConnectionBegin	pfc_PreLogonDlg
pfc_ConnectionEnd	pfc_PreSplash

pfc_Exit	pfc_SystemError
pfc_Idle	

Constructor

Description

Initializes the iapp_object and ienv_object instance variables.

Usage

Initialize application-wide instance variables in this event. Depending on your application's needs, you initialize the following:

Instance variable	Initialization function
is_appinifile	of_SetAppIniFile
is_appkey	of_SetAppKey
is_appinifile	of_SetAppIniFile
is_helpfile	of_SetHelpFile
is_userinifile	of_SetUserIniFile
is_userkey	of_SetUserKey
is_version	of_SetVersion

Examples

This example initializes selected instance variables:

```
of_SetAppIniFile("c:\eis\eisapp.ini")
of_SetUserIniFile("c:\eis\eisuser.ini")
of_SetHelpFile("c:\eis\eis.hlp")
of_SetVersion("1.1")
```

Destructor

Description

Destroys the service objects associated with an instance of n_cst_appmanager.

Usage

Use this event to destroy any other user objects associated with n_cst_appmanager.

Examples

This example checks for the existence of the inv_app_logic custom class user object and destroys it if an instance exists:

```
IF IsValid(gnv_app.inv_app_logic) THEN
   DESTROY gnv_app.inv_app_logic
END IF
```

pfc_Close

Description

Application manager Close event.

Usage

To use this event:

1 Call pfc_Close from the application object's Close event.

2 Add application close logic to pfc_Close.

Examples

This example disconnects from the database:

```
Integer   li_return

IF IsValid itr_app_trans THEN
   li_return = itr_app_trans.of_Disconnect()
   IF li_return <> 0 THEN
      MessageBox("Disconnect Error", &
         itr_app_trans.SqlErrText)
   END IF
END IF
```

pfc_ConnectionBegin

Description

Application manager ConnectionBegin event. The ConnectionBegin event is triggered by a distributed application when a client application attempts to establish a connection to a server application. This event is triggered only in a server application running in a distributed computing environment.

Syntax

instancename.EVENT **pfc_ConnectionBegin** (*userid*, *password*, *connectstring*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
userid	String specifying the user ID. From within the pfc_ConnectionBegin event, access this value through the *as_userid* argument
password	String specifying the password. From within the pfc_ConnectionBegin event, access this value through the *as_password* argument
connectstring	String containing the connectstring. From within the pfc_ConnectionBegin event, access this value through the *as_connectstring* argument

Return value

ConnectPrivilege. Returns one of the following:

ConnectPrivilege!
ConnectWithAdminPrivilege!
NoConnectPrivilege!

Usage

To use this event:

1 Call pfc_ConnectionBegin from the application object's ConnectionBegin event, passing the *userid*, *password*, and *connectstring* arguments.

2 Add connection logic to pfc_ConnectionBegin, returning the appropriate connect privilege.

3 In the application object's ConnectionBegin event, return the connect privilege.

FOR INFO For more information on distributed computing, see *Application Techniques*.

Examples

This example returns the client's connection privilege:

```
CHOOSE CASE as_userid
   CASE "adm"
      Return ConnectWithAdminPrivilege!
   CASE "ctl","mmw","jdb"
      Return ConnectPrivilege!
   CASE ELSE
      Return NoConnectPrivilege!
END CHOOSE
```

pfc_ConnectionEnd

Description

Application manager ConnectionEnd event. The ConnectionEnd event is triggered by a distributed application when a client application's connection is terminated. This event is triggered only in a server application running in a distributed computing environment.

Usage

To use this event:

1 Call pfc_ConnectionEnd from the application object's ConnectionEnd event.

2 Add logic to pfc_ConnectionEnd.

FOR INFO For more information on distributed PowerBuilder, see *Application Techniques*.

Examples

This example disconnects from the database:

```
Integer   li_return

IF IsValid itr_app_trans THEN
   li_return = itr_app_trans.of_Disconnect()
END IF
```

pfc_Exit

Description

Event to close the frame window, shutting down the application.

Usage

Add code to this event to perform the appropriate end-of-application processing (for example, closing the frame window.)

Examples

This example closes the frame window:

```
w_frame   lw_frame

lw_frame = of_GetFrame()
Close(lw_frame)
```

pfc_Idle

Description

Application manager Idle event.

Usage

To use this event:

1 Call pfc_Idle from the application object's Idle event.

2 Add application idle logic to pfc_Idle.

Examples

This example triggers the pfc_Exit event to shut down the application when the pfc_Idle event is triggered:

```
this.Event pfc_Exit()
```

pfc_Logon

Description

Application manager logon event. This event is called by the w_logon dialog box when the user clicks OK.

Syntax

instancename.EVENT **pfc_Logon** (*userid*, *password*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
userid	String specifying the user ID. From within the pfc_Logon event, access this value through the *as_userid* argument
password	String specifying the password. From within the pfc_Logon event, access this value through *as_password* argument

Return value

Integer. Returns 1 if the logon succeeds and -1 if the logon fails.

Usage

Add code to use this event to log on to the database using the passed user ID and password.

Examples

This example takes the passed user ID and password and uses them to log on to the database (assuming an itr_default transaction object):

```
itr_default.of_SetUser(as_userid, as_password)
IF itr_default.of_Connect() >= 0 THEN
   Return 1
ELSE
   Return -1
END IF
```

pfc_Open

Description

Application manager Open event.

Syntax

instancename.EVENT **pfc_Open** (*commandline*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
commandline	String containing the command-line options specified when the program was invoked

Usage

To use this event:

1 Call pfc_Open from the Application object's Open event (after creating gnv_app).

2 Add application open logic to pfc_Open.

Examples

This example enables transaction registration and opens the w_test_frame window:

```
// Enable transaction registration services
this.of_SetTrRegistration(TRUE)

// Register SQLCA
inv_trregistration.of_Register(SQLCA)

// Now open the frame
Open(w_test_frame)
```

pfc_PreAbout

Description

Event to control display of the application's About dialog box. This event copies application, logo, version, and copyright information to the n_cst_aboutattrib object, which is used in the About dialog box.

Syntax

instancename.EVENT **pfc_PreAbout** (*attributes*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
attributes	N_cst_aboutattrib variable into which the event places application, logo, version, and copyright information. This argument is accessed through the *anv_aboutattrib* argument (passed by reference)

Usage

This event is triggered when the user selects Help>About in a menu that descends from the PFC m_master menu.

Examples

This example is from the of_About function:

```
n_cst_aboutattrib   lnv_aboutattrib

this.Event pfc_PreAbout(lnv_aboutattrib)
Return OpenWithParm (w_about, lnv_aboutattrib)
```

pfc_PreLogonDlg

Description

Event to control display of the w_logon dialog box. This event copies information to the n_cst_logonattrib structure object.

Syntax

instancename.EVENT **pfc_PreLogonDlg** (*attributes*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Argument	Description
attributes	N_cst_logonattrib variable into which the event places user ID, application, and logo information. This argument is accessed through the *anv_logonattrib* argument (passed by reference)

Usage

This event is called by the of_LogonDlg function. Extend or update *attributes* to control display of the w_logon dialog box.

Examples

This example is from the of_LogonDlg function:

```
Integer   li_rc
String    ls_userid
n_cst_logonattrib   lnv_logonattrib

this.Event pfc_PreLogonDlg(lnv_logonattrib)
...
```

pfc_PreSplash

Description

Event to control display of the application's splash screen.

Syntax

instancename.Event **pfc_PreSplash** (*attributes*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
attributes	N_cst_splashattrib variable into which the event places application, logo, version, and copyright information. This argument is accessed through the *anv_splashattrib* argument (passed by reference)

Usage

This event is called by the of_Splash function.

This event copies application, logo, version, and copyright information to the n_cst_splashattrib object, which is used in the splash window.

Examples

This example is from the of_Splash function:

```
n_cst_splashattrib   lnv_splashattrib

lnv_splashattrib.ii_secondsvisible = &
   ai_secondsvisible
this.Event pfc_PreSplash(lnv_splashattrib)

OpenWithParm (w_splash, lnv_splashattrib)
```

pfc_SystemError

Description

Application manager SystemError event.

Usage

To use this event:

1 Call pfc_SystemError from the application object's SystemError event.

2 (Optional) Add system error logic to pfc_SystemError.

Examples

This example displays information from the Error object:

```
MessageBox("System Error", &
    "Error Number: " + String(Error.Number) &
    + "~r~n" &
    + "Message: " + Error.Text + "~r~n" &
    + "Where: " + Error.WindowMenu + "~r~n" &
    + "Object: " + Error.Object + "~r~n" &
    + "Event: " + Error.ObjectEvent + "~r~n" &
    + "Line: " + String(Error.Line) )
```

Functions

N_cst_appmanager includes precoded object functions:

of_About	of_SetCopyright
of_GetAppIniFile	of_SetDebug
of_GetAppKey	of_SetDwCache
of_GetCopyright	of_SetError
of_GetFrame	of_SetFrame
of_GetHelpFile	of_SetHelpFile
of_GetLogo	of_SetLogo
of_GetMicrohelp	of_SetMicrohelp
of_GetUserID	of_SetMRU
of_GetUserIniFile	of_SetSecurity
of_GetUserKey	of_SetTrRegistration
of_GetVersion	of_SetUserID
of_IsRegistryAvailable	of_SetUserIniFile
of_LogonDlg	of_SetUserKey
of_SetAppIniFile	of_SetVersion
of_SetAppKey	of_Splash
of_SetAppPreference	

of_About

Description

Displays the w_about dialog box.

Access

Public

Syntax

instancename.**of_About ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

PFC calls this function when the user selects Help>About from the menu bar of a menu descended from the PFC m_master menu.

Examples

This example is from the m_master Help>About Clicked event:

```
IF IsValid (gnv_app) THEN
   gnv_app.of_About()
END IF
```

of_GetAppIniFile

Description

Retrieves the name of the application INI file as established by the of_SetAppIniFile function.

Access

Public

Syntax

instancename.**of_GetAppIniFile ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the is_appinifile instance variable, which contains the name of the application-level INI file (specified using the of_SetAppIniFile function).

Usage

Call this function in profile functions (such as ProfileString) to access information in the application INI file. The application INI file typically contains application wide information, such as the database connection profile.

Examples

This example uses the of_GetAppIniFile function to retrieve database connection information:

```
SQLCA.dbms=ProfileString &
   (gnv_app.of_GetAppIniFile(), &
```

```
        "Database","DBMS"," ")
SQLCA.database=ProfileString &
    (gnv_app.of_GetAppIniFile(), &
        "Database","Database"," ")
SQLCA.servername=ProfileString &
    (gnv_app.of_GetAppIniFile(), &
        "Database","ServerName"," ")
SQLCA.logid=ProfileString &
    (gnv_app.of_GetAppIniFile(), &
        "Database","LogID"," ")
SQLCA.logpass=ProfileString &
    (gnv_app.of_GetAppIniFile(), &
        "Database","LogPassword"," ")
```

of_GetAppKey

Description

Retrieves the registry key for application information as established by the of_SetAppKey function.

Access

Public

Syntax

instancename.**of_GetAppKey ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the is_appkey instance variable.

Examples

This example calls the of_GetAppKey function:

```
String    ls_appkey

ls_appkey = gnv_app.of_GetAppKey()
MessageBox("Application", &
    "Application key is:~r~n" + ls_appkey)
```

of_GetCopyright

Description

Retrieves the copyright statement as established by the of_SetCopyright function.

Access

Public

Syntax

instancename.**of_GetCopyright ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the is_copyright instance variable.

Usage

Call this function to access the application's copyright statement, which is contained in the is_copyright instance variable.

Examples

This example calls the of_GetCopyright function:

```
String    ls_copyright

ls_copyright = gnv_app.of_GetCopyright()
MessageBox("Application", &
    "Copyright is:~r~n" + ls_copyright)
```

of_GetFrame

Description

Retrieves the active frame window as established by the of_SetFrame function.

Access

Public

Syntax

instancename.**of_GetFrame ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

W_frame. Returns the application's active frame window.

Usage

Call this function in PowerScript functions that require the frame window.

Examples

This example uses the of_GetFrame function in a SetMicroHelp function:

```
gnv_app.of_GetFrame().SetMicroHelp("Connecting...")
```

of_GetHelpFile

Description

Retrieves the name of the application's online Help file as established by the of_SetHelpFile function.

Access

Public

Syntax

instancename.**of_GetHelpFile ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the name of the application's online Help file.

Usage

Call this function instead of hard coding the online Help filename in the PowerScript ShowHelp function.

Examples

This example uses the of_GetHelpFile function as part of a PowerScript ShowHelp function that displays the main Help topic:

```
ShowHelp(gnv_app.of_GetHelpFile(), Index!)
```

of_GetLogo

Description

Retrieves the name of the logo as established by the of_SetLogo function.

Access

Public

Syntax

instancename.**of_GetLogo ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the is_logo instance variable.

Examples

This example calls the of_GetLogo function:

```
String    ls_logo

ls_logo = gnv_app.of_GetLogo()
MessageBox("Application", &
    "Logo is:~r~n" + ls_logo)
```

of_GetMicrohelp

Description

Retrieves the application's current MicroHelp behavior as established by the of_SetMicrohelp function.

Access

Public

Syntax

instancename.**of_GetMicrohelp ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

Boolean. Returns TRUE if MicroHelp is enabled and FALSE if it is not.

Examples

This example calls the of_GetMicrohelp function:

```
IF gnv_app.of_GetMicrohelp() THEN
   MessageBox("Application", &
      "Microhelp is enabled")
ELSE
   MessageBox("Application", &
      "Microhelp is disabled")
END IF
```

of_GetUserID

Description

Retrieves the current user's ID as established by the of_SetUserID function.

Access

Public

Syntax

instancename.**of_GetUserID** ( )

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the ID of the current user, which is contained in the is_userid instance variable.

Examples

This example uses the of_GetUserID function to populate the st_user static text field:

```
st_version.text = gnv_app.of_GetVersion()
st_user.text = gnv_app.of_GetUserID()
```

of_GetUserIniFile

Description

Retrieves the name of the user INI file as established by the of_SetUserIniFile function.

Access

Public

Syntax

instancename.**of_GetUserIniFile** ( )

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the is_userinifile instance variable, which contains the name of the user-specific INI file.

Usage

Call this function in profile functions (such as ProfileString) to access information in the user INI file.

The user INI file typically contains user-specific information, such as toolbar settings and window position.

Examples

This example calls the of_GetUserIniFile function to name the user INI file in the n_cst_winsrv_preference of_Save function:

```
this.inv_preference.of_Save &
    (gnv_app.of_GetUserIniFile(), "Position")
```

of_GetUserKey

Description

Retrieves the registry key for user information as established by the of_SetUserKey function.

Access

Public

Syntax

instancename.**of_GetUserKey ( )**

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the is_userkey instance variable.

Examples

This example calls the of_GetUserKey function:

```
String    ls_userkey

ls_userkey = gnv_app.of_GetUserKey()
MessageBox("Application", &
    "User key is:~r~n" + ls_userkey)
```

of_GetVersion

Description

Retrieves the current version as established by the of_SetVersion function.

Access

Public

Syntax

instancename.**of_GetVersion** ( )

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

String. Returns the application version, which is contained in the is_version instance variable.

Examples

This example uses the of_GetVersion function to populate the st_version static text field:

```
st_version.text = gnv_app.of_GetVersion()
st_user.text = gnv_app.of_GetUserID()
```

of_IsRegistryAvailable

Description

Reports whether the workstation supports the registry.

Access

Public

Syntax

instancename.**of_IsRegistryAvailable** ( )

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

Boolean. Returns TRUE if the workstation is running Windows 95 or Windows NT and FALSE if it is not.

Usage

You can use this function to add cross-platform support to an application, writing information to either the registry or an INI file, depending on the platform.

If you never want to use the registry
PFC uses this function internally to determine whether to access the registry or an INI file. If your application never uses the registry, override this function's extension-level descendant to always return FALSE.

Examples

This example calls the of_IsRegistryAvailable function:

```
String    ls_setting
Integer   li_return
```

```
IF gnv_app.of_IsRegistryAvailable() THEN
   registryGet(gnv_app.of_GetUserKey(), &
      "ToolbarSettings", ls_setting)
   IF ls_setting = "" THEN ls_setting = "TRUE"
ELSE
   ls_setting = &
      ProfileString(gnv_app.of_GetUserIniFile(), &
      "ToolbarSettings", "Visible", "TRUE")
END IF
MessageBox("Application", "Visible setting is " &
   + ls_setting)
```

of_LogonDlg

Description

Displays the w_logon dialog box and then calls the pfc_Logon event, allowing you to log the user on to the database.

Access

Public

Syntax

instancename.**of_LogonDlg** ( )

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)

Return value

Integer. Returns 1 if the function succeeds, 0 if the user cancels out of the dialog box, and -1 if an error occurs.

Usage

If you call this function, you must also add code to pfc_Logon to perform the actual database logon.

Examples

This example is from a frame window's pfc_Open event:

```
of_Splash (1)
Open(w_frame)
IF of_LogonDlg() = 0 THEN
   this.Event pfc_Exit()
END IF
```

of_SetAppIniFile

Description

Specifies the application INI file. The application INI file contains application-wide settings, such as database connection information. N_cst_appmanager maintains this setting in the is_appinifile instance variable.

Access

Public

Syntax

instancename.**of_SetAppIniFile** (*inifilename*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
inifilename	String specifying the fully qualified name of the application INI file

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You should call this function at application startup, typically in the n_cst_appmanager Constructor event.

Your application accesses the application INI filename through the of_GetAppIniFile function.

Examples

This example calls the of_SetAppIniFile function in the n_cst_appmanager Constructor event (since it's called from within the object, it doesn't require the gnv_app prefix):

```
of_SetAppIniFile("c:\eis\eisapp.ini")
```

of_SetAppKey

Description

Specifies the registry key for application information.

Access

Public

Syntax

instancename.**of_SetAppKey** (*registrykey*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
registrykey	String containing the registry key

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The application registry key is contained in the is_appkey instance variable.

Examples

This example calls the of_SetAppKey function:

```
String   ls_appkey

ls_appkey = "HKEY_CURRENT_USER\Software\EISAPP"
gnv_app.of_SetAppKey(ls_appkey)
```

```
gnv_app.of_GetFrame().SetMicroHelp &
   ("Application key set successfully")
```

of_SetAppPreference

Description

Enables or disables the application preference service. The application preference service is implemented through the n_cst_apppreference user object. You reference this instance of n_cst_apppreference through the inv_apppref instance variable.

Access

Public

Syntax

instancename.**of_SetAppPreference** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the application preference service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example enables the security service:

```
gnv_app.of_SetAppPreference(TRUE)
```

of_SetCopyright

Description

Sets the copyright statement that displays in the w_about dialog box.

Access

Public

Syntax

instancename.**of_SetCopyright** (*copyrighttext*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
copyrighttext	String specifying the copyright text

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetCopyright function:

```
String    ls_copyright
```

```
ls_copyright = &
   "Acme Software, Inc. All rights reserved"
gnv_app.of_SetCopyright(ls_copyright)
```

of_SetDebug

Description

Enables or disables the debugging service. The debugging service is implemented through the n_cst_debug user object. You reference this instance of n_cst_debug through the inv_debug instance variable.

Access

Public

Syntax

instancename.**of_SetDebug** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the PFC debugging service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetDebug function:

```
gnv_app.of_SetDebug(TRUE)
```

of_SetDwCache

Description

Enables or disables the data caching service. The data caching service is implemented through the n_cst_dwcache user object. You reference this instance of n_cst_dwcache through the inv_dwcache instance variable.

Access

Public

Syntax

instancename.**of_SetDwCache** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the data caching service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example enables the data caching service:

```
gnv_app.of_SetDwCache(TRUE)
```

of_SetError

Description

Enables or disables the error service. The error service is implemented through the n_cst_error user object. You reference this instance of n_cst_error through the inv_error instance variable.

Access

Public

Syntax

instancename.**of_SetError** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app).
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the error service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example enables the error service:

```
gnv_app.of_SetError(TRUE)
```

of_SetFrame

Description

Assigns the active frame window to the iw_frame instance variable.

Access

Public

Syntax

instancename.**of_SetFrame** (*framewindow*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
framewindow	W_frame variable specifying the frame window

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The w_frame Activate event calls this function to establish the iw_frame instance variable.

Examples

This example is from the w_frame Activate event and establishes the active frame window:

```
IF IsValid (gnv_app) THEN
   gnv_app.of_SetFrame(this)
END IF
```

of_SetHelpFile

Description

Specifies the application's Help file. N_cst_appmanager maintains this setting in the is_helpfile instance variable.

Access

Public

Syntax

instancename.**of_SetHelpFile** (*filename*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
filename	String specifying the fully qualified name of the online Help file

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You should call this function at application startup, typically in the n_cst_appmanager Constructor event. Your application accesses the Help filename through the of_GetHelpFile function.

Examples

This example calls the of_SetHelpFile function in the n_cst_appmanager Constructor event (since it's called from within the object, it doesn't require the gnv_app prefix):

```
of_SetHelpFile("c:\eis\eisapp.hlp")
```

of_SetLogo

Description

Specifies the name of the logo bitmap that displays in the w_about dialog box.

Access

Public

Syntax

instancename.**of_SetLogo** (*logoname*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
logoname	String specifying the name of the bitmap file containing the logo. This must either be a fully qualified filename or reside in a directory on the system path

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetLogo function.

```
String    ls_logo

ls_logo = "bubbles.bmp"
gnv_app.of_SetLogo(ls_logo)
```

of_SetMicrohelp

Description

Enables or disables automatic MicroHelp display.

Access

Public

Syntax

instancename.**of_SetMicrohelp** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) default MicroHelp display

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMicrohelp function:

```
gnv_app.of_SetMicrohelp(TRUE)
```

of_SetMRU

Description

Enables or disables the most recently used object service. The most recently used object service is implemented through the n_cst_mru user object. You reference this instance of n_cst_mru through the inv_mru instance variable.

Access

Public

Syntax

instancename.**of_SetMRU** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the most recently used object service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example enables the most recently used object service:

```
gnv_app.of_SetMRU(TRUE)
```

of_SetSecurity

Description

Enables or disables the security service. The security service is implemented through the n_cst_security user object. You reference this instance of n_cst_security through the inv_security instance variable.

Access

Public

Syntax

instancename.**of_SetSecurity** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the security service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example enables the security service:

```
gnv_app.of_SetSecurity(TRUE)
```

of_SetTrRegistration

Description

Enables or disables the transaction registration service. The transaction registration service is implemented through the n_cst_trregistration user object. You reference this instance of through the inv_trregistration instance variable.

Access

Public

Syntax

instancename.**of_SetTrRegistration** (*boolean*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) the transaction registration service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example enables the transaction registration service:

```
gnv_app.of_SetTrRegistration(TRUE)
```

of_SetUserID

Description

Saves the passed user name in the is_userid instance variable.

Access

Public

Syntax

instancename.**of_SetUserID** (*userid*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
userid	String specifying the user id

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to establish the user ID of the current user.

Examples

This example sets the current user equal to the user ID entered in the sle_user SingleLineEdit:

```
IF sle_user.text <> "" THEN
   gnv_app.of_SetUserID(sle_user.text)
END IF
```

of_SetUserIniFile

Description

Specifies the user INI file. The user INI file contains user-specific settings, such as window position information. N_cst_appmanager maintains this setting in the is_userinifile instance variable.

Access

Public

Syntax

instancename.**of_SetUserIniFile** (*inifilename*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
inifilename	String specifying the fully qualified name of the user INI file

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You should call this function at application startup, typically in the n_cst_appmanager Constructor event.

Your application accesses the application INI filename through the of_GetUserIniFile function.

Examples

This example calls the of_SetUserIniFile function in the n_cst_appmanager Constructor event (since it's called from within the object, it doesn't require the gnv_app prefix):

```
of_SetUserIniFile("c:\eis\eisuser.ini")
```

of_SetUserKey

Description

Specifies the registry key for user information.

Access

Public

Syntax

instancename.**of_SetUserKey** (*registrykey*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
registrykey	String containing the registry key

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to set the user's registry key, which is contained in the is_userkey instance variable.

Examples

This example calls the of_SetUserKey function:

```
String   ls_userkey

ls_userkey = "HKEY_CURRENT_USER\Software\EISUSER"
gnv_app.of_SetUserKey(ls_userkey)
gnv_app.of_GetFrame().SetMicroHelp &
    ("User key set successfully")
```

of_SetVersion

Description

Specifies the application version. N_cst_appmanager maintains this setting in the is_version instance variable.

Access

Public

Syntax

instancename.**of_SetVersion** (*version*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
version	String specifying the version

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You should call this function at application startup, typically in the n_cst_appmanager Constructor event.

Your application accesses the application version through the of_GetVersion function.

Examples

This example calls the of_SetVersion function in the n_cst_appmanager Constructor event (since it's called from within the object, it doesn't require the gnv_app prefix):

```
of_SetVersion("Version 2.1B")
```

of_Splash

Description

Displays the w_splash window for the specified number of seconds.

Access

Public

Syntax

instancename.**of_Splash** (*secondsvisible*)

Argument	Description
instancename	Global variable of type n_cst_appmanager (must be gnv_app)
secondsvisible	Integer specifying the number of seconds to display the w_splash window

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

To use this function:

1 (Optional) Modify the w_splash extension-level window to display site-specific information.

2 (Optional) Extend the pfc_PreSplash event to dynamically modify w_splash.

3 In the n_cst_appmanager Constructor event, call functions to specify application, copyright, logo, and version information.

4 In the pfc_Open event, call this function just before opening the main window.

Examples

This example calls the of_Splash function:

```
of_Splash(1)
Open(w_tut_frame)
IF of_LogonDlg() = 0 THEN
   this.Event pfc_Exit()
END IF
```

n_cst_apppreference

Description

Application preference service. When enabled, you can use n_cst_apppreference to save and restore application and user information using either an INI file or the Windows registry. N_cst_apppreference saves the following *application* information:

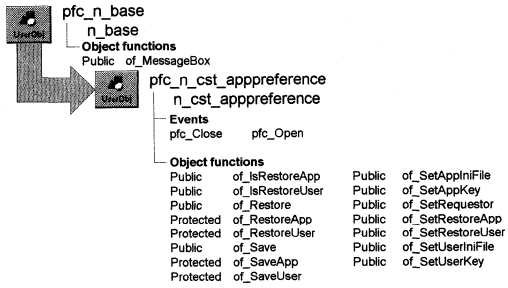
User key
MicroHelp
Help file
Version
Logo bitmap
Copyright notice
DDETimeOut property
DisplayName property
DWMessageTitle property
MicrohelpDefault property
RightToLeft property
ToolbarFrameTitle property
ToolbarPopMenuText property
ToolbarSheetTitle property
ToolbarUserControl property

N_cst_apppreference can also save the following *user* information:

ToolbarText property
ToolbarTips property
User ID

N_cst_apppreference automatically loads settings when the application opens and stores them when the application closes. This information is stored in either the registry (available on Windows 95 and Windows NT only) or an INI file (available on all platforms).

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_appatrib
n_cst_conversion
n_cst_inifile
n_cst_string

Usage

Use this object to save and restore application information.

To use n_cst_apppreference:

1 Enabled n_cst_apppreference by calling the n_cst_appmanager of_SetAppPreference function (this example is from an n_cst_appmanager Constructor event):

```
this.of_SetAppPreference(TRUE)
```

2 Specify the platform-specific repository for application preferences. This example from an application manager Constructor event saves application preferences in the registry or INI file, depending on the execution platform. It assumes you've already established n_cst_appmanager specifications for application key, user key, application INI file, and user INI file:

```
IF this.of_IsRegistryAvailable() THEN
   this.inv_apppref.of_SetAppKey &
     (this.of_GetAppKey())
   this.inv_apppref.of_SetUserKey &
     (this.of_GetUserKey())
ELSE
   this.inv_apppref.of_SetAppINIFile &
     (this.of_GetAppINIFile())
   this.inv_apppref.of_SetUserINIFile &
     (this.of_GetUserINIFile())
END IF
```

3 Specify the types of information to save by calling the of_SetRestoreApp and of_SetRestoreUser functions:

```
this.inv_apppref.of_SetRestoreApp(TRUE)
this.inv_apppref.of_SetRestoreUser(TRUE)
```

See also

n_cst_appmanager
n_cst_winsrv_preference

Instance variables

N_cst_apppreference includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_restoreapp	Controls whether the service saves and restores application-wide settings	Boolean	Protected	Set with of_SetRestoreApp
ib_restoreuser	Controls whether the service saves and restores user-specific settings	Boolean	Protected	Set with of_SetRestoreUser
inv_appmanager	Reference to application manager	n_cst_appmanager	Protected	Used to access application information
is_appinifile	INI file that contains application information	String	Protected	Set with of_SetAppINIFile
is_appkey	Registry key that contains application information	String	Protected	Set with of_SetAppKey
is_appprefsection	INI file section or registry key for application information	String	Protected	Internal. Default is AppPref
is_userinifile	INI file that contains user information	String	Protected	Set with of_SetUserINIFile
is_userkey	Registry key that contains application information	String	Protected	Set with of_SetUserKey
is_userpref	INI file or registry key for user information	String	Protected	Internal. Default is AppUserPref

Events

N_cst_apppreference includes precoded events:

pfc_Close
pfc_Open

pfc_Close

Description

Saves application and user properties when the application closes.

Syntax

*application.instancename.*EVENT**pfc_Close** ( )

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

This event executes when the application closes; it is called by the n_cst_appmanager pfc_Close event if n_cst_apppreference is enabled. It saves application and user properties as specified by the of_SetRestoreApp and of_SetRestoreUser functions.

pfc_Open

Description

Restores application and user properties when the application opens.

Syntax

*application.instancename.*EVENT**pfc_Open** (*commandline*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
commandline	String containing the command-line options specified when the program was invoked

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

This event executes when the application opens; it is called by the n_cst_appmanager pfc_Open event if n_cst_apppreference is enabled. It restores application and user properties as specified by the of_SetRestoreApp and of_SetRestoreUser functions, which you typically call in the n_cst_appmanager Constructor event.

Functions

N_cst_apppreference contains precoded object functions:

of_IsRestoreApp	of_SetAppIniFile
of_RestoreUser	of_SetAppKey
of_Restore	of_SetRequestor
of_RestoreApp	of_SetRestoreApp
of_RestoreUser	of_SetRestoreUser
of_Save	of_SetUserIniFile
of_SaveApp	of_SetUserKey
of_SaveUser	

of_IsRestoreApp

Description

Reports whether n_cst_apppreference saves and restores application settings.

Access

Public

Syntax

application.instancename.**of_IsRestoreApp** ( )

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)

Return value

Boolean. Returns TRUE if the service saves and restores application settings and FALSE if it does not.

Examples

This example calls the of_IsRestoreApp function:

```
n_cst_conversion lnv_conversion
Boolean lb_appinfo

lb_appinfo = &
gnv_app.inv_apppref.of_IsRestoreApp()
MessageBox("App Preferences", &
  "Save app info is " &
    + lnv_conversion.of_String(lb_appinfo))
```

of_IsRestoreUser

Description

Reports whether n_cst_apppreference saves and restores user settings.

Access

Public

Syntax

application.instancename.**of_IsRestoreUser** ( )

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)

Return value

Boolean. Returns TRUE if the service saves and restores user settings and FALSE if it does not.

Examples

This example calls the of_IsRestoreUser function:

```
n_cst_conversion lnv_conversion
Boolean lb_userinfo

lb_userinfo = &
gnv_app.inv_apppref.of_IsRestoreUser()
MessageBox("App Preferences", &
  "Save user info is " &
    + lnv_conversion.of_String(lb_userinfo))
```

of_Restore

Restores application and user settings from the registry or an INI file:

To	Use
Restore settings from the registry	Syntax 1
Restore settings from an INI file	Syntax 2

Other syntaxes
There are two additional protected versions of the of_Restore function. These are for internal use.

Syntax 1

Restore settings from the registry

Description

Restores application and user settings from the registry.

Access

Public

Syntax

application.instancename.**of_Restore** (*processapp*, *registrykey*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
processapp	Boolean indicating whether to restore application settings (TRUE) or user settings (FALSE)
registrykey	String specifying the registry key in which to restore settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If n_cst_apppreference is enabled, the n_cst_apppreference pfc_Open event calls this function automatically.

This syntax is for the Windows 95 and Windows NT platforms only. Use Syntax 2 to restore settings from an INI file.

Examples

This example is from the pfc_Open event:

```
...
IF ib_restoreapp THEN
  IF (is_appkey <> "") THEN
    li_rc = of_Restore (TRUE, is_appkey)
  ELSEIF (is_appinifile <> "") THEN
    li_rc = of_Restore(TRUE, &
    is_appinifile, is_apppref)
  ELSE
    Return 0
  END IF
END IF
...
```

Syntax 2

Restore settings from an INI file

Description

Restores application and user settings from an INI file.

Access

Public

Syntax

application.instancename.**of_Restore** (*processapp*, *inifile*, *inisection*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_appref)
processapp	Boolean indicating whether to restore application settings (TRUE) or user settings (FALSE)
inifile	String specifying the INI file from which to restore settings
inisection	String specifying the INI file section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If n_cst_apppreference is enabled, the n_cst_apppreference pfc_Open event calls this function automatically.

This syntax is for all platforms. Use Syntax 1 to restore settings from the registry.

Examples

This example is from the pfc_Open event:

```
...
IF ib_restoreapp THEN
  IF (is_appkey <> "") THEN
    li_rc = of_Restore(TRUE, is_appkey)
  ELSEIF (is_appinifile <> "") THEN
    li_rc = of_Restore (TRUE, &
      is_appinifile, is_apppref )
  ELSE
    Return 0
  END IF
END IF
...
```

of_RestoreApp

Description

Restores application information from either the registry or an INI file.

Access

Protected

Syntax

*application.instancename.***of_RestoreApp** (*useregistry, keyorinifile, section*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
keyorinifile	String specifying either the registry key or the INI file from which to restore settings
section	String specifying the section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Restore function:

```
...
IF ab_processapp THEN
  li_rc =  of_RestoreApp (ab_useregistry, &
    as_keyorini, as_inisection)
ELSE
  li_rc = of_RestoreUser(ab_useregistry, &
    as_keyorini, as_inisection
END IF
...
```

of_RestoreUser

Description

Restores user information from either the registry or an INI file.

Access

Protected

Syntax

*application.instancename.***of_RestoreUser** (*useregistry, keyorinifile,*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
keyorinifile	String specifying either the registry key or the INI file from which to restore settings

Argument	Description
section	String specifying the section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Restore function:

```
...
IF ab_processapp THEN
  li_rc = of_RestoreApp(ab_useregistry, &
  as_keyorini, as_inisection)
ELSE
  li_rc = of_RestoreUser (ab_useregistry, &
  as_keyorini, as_inisection)
END IF
...
```

of_Save

Saves window, toolbar, and menu settings in the registry or an INI file. There are two syntaxes:

To	Use
Save settings in the registry	Syntax 1
Save settings in an INI file	Syntax 2

Other syntaxes
There are two additional protected versions of the of_Save function. These are for internal use.

Syntax 1

Save settings in the registry

Description

Saves application and user settings in the registry.

Access

Public

Syntax

application.instancename.**of_Save** (*processapp*, *registrykey*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
processapp	Boolean indicating whether to save application settings (TRUE) or user settings (FALSE)
registrykey	String specifying the registry key in which to save settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If n_cst_apppreference is enabled, the n_cst_apppreference pfc_Close event calls this function automatically.

This syntax is for the Windows 95 and Windows NT platforms only. Use Syntax 2 to save settings in an INI file.

Examples

This example is from the pfc_Close event:

```
...
IF ib_restoreapp THEN
  IF (is_appkey <> "") THEN
    li_rc = of_Save(TRUE, is_appkey)
  ELSEIF (is_appinifile <> "") THEN
    li_rc = of_Save(TRUE, is_appinifile, &
      is_appprefsection)
  ELSE
    Return 0
  END IF
END IF
...
```

Syntax 2

Save settings in an INI file

Description

Saves application and user settings in an INI file.

Access

Public

Syntax

*application.instancename.***of_Save** (*processapp, inifile, section*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
processapp	Boolean indicating whether to save application settings (TRUE) or user settings (FALSE)
inifile	String specifying the INI file in which to save settings
section	String specifying the INI file section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If n_cst_apppreference is enabled, the n_cst_apppreference pfc_Close event calls this function automatically.

This syntax is for all platforms. Use Syntax 1 to save settings in the registry.

Examples

This example is from the pfc_Close event:

```
...
IF ib_restoreapp THEN
  IF (is_appkey <> "") THEN
    li_rc = of_Save(TRUE, is_appkey)
  ELSEIF (is_appinifile <> "") THEN
    li_rc = of_Save(TRUE, is_appinifile, &
      is_appprefsection)
  ELSE
    Return 0
  END IF
END IF
...
```

of_SaveApp

Description

Saves application information in either the registry or an INI file.

Access

Protected

Syntax

*application.instancename.***of_SaveApp** (*useregistry, keyorinifile, section*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is apppref)

Argument	Description
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
keyorinifile	String specifying the registry key or INI file in which to save settings
section	String specifying the INI file section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_Save function:

```
...
IF ab_ProcessApp THEN
  li_rc = of_SaveApp (ab_useregistry, &
    as_keyorini, as_inisection)
ELSE
  li_rc = of_SaveUser(ab_useregistry, &
    as_keyorini, as_inisection)
END IF
...
```

of_SaveUser

Description

Saves user information in either the registry or an INI file.

Access

Protected

Syntax

*application.instancename.***of_SaveUser** (*useregistry, keyorinifile, section*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
keyorinifile	String specifying the registry key or INI file in which to save settings
section	String specifying the registry key or INI file section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_SaveUser function:

```
...
IF ab_ProcessApp THEN
li_rc = of_SaveApp(ab_useregistry, &
as_keyorini, as_inisection)
ELSE
li_rc = of_SaveUser(ab_useregistry, &
as_keyorini, as_inisection)
END IF
...
```

of_SetAppIniFile

Description

Specifies the name of the application INI file, which contains application preferences. N_cst_apppreference uses this file by default on 16-bit Windows, UNIX, and the Macintosh.

Access

Public

Syntax

application.instancename.**of_SetAppIniFile** (*inifilename*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
inifilename	String specifying the fully qualified name of the application INI file

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The INI file you set for application preferences is typically the same as the n_cst_appmanager application INI file.

Examples

This example from an n_cst_appmanager Constructor event calls the of_SetAppINIFile function:

```
...
IF this.of_IsRegistryAvailable() THEN
  this.inv_apppref.of_SetAppKey &
```

```
    (this.of_GetAppKey())
  this.inv_apppref.of_SetUserKey &
    (this.of_GetUserKey())
ELSE
  this.inv_apppref.of_SetAppINIFile &
    (this.of_GetAppINIFile())
  this.inv_apppref.of_SetUserINIFile &
    (this.of_GetUserINIFile())
END IF
...
```

of_SetAppKey

Description

Specifies the name of the registry key for application information. N_cst_apppreference uses this key by default on Windows 95 and Windows NT.

Access

Public

Syntax

application.instancename.**of_SetAppKey** (*registrykey*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
registrykey	String containing the registry key

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The registry key you set for application information is typically the same as the n_cst_appmanager application key.

Examples

This example calls the of_SetAppKey function:

```
...
IF this.of_IsRegistryAvailable() THEN
  this.inv_apppref.of_SetAppKey &
    (this.of_GetAppKey())
  this.inv_apppref.of_SetUserKey &
    (this.of_GetUserKey())
ELSE
  this.inv_apppref.of_SetAppINIFile &
    (this.of_GetAppINIFile())
```

```
    this.inv_apppref.of_SetUserINIFile &
      (this.of_GetUserINIFile())
END IF
...
```

of_SetRequestor

Description — Associates n_cst_appmanager with n_cst_apppreference.

Access — Public

Syntax — *application.instancename*.**of_SetRequestor** (*requestor*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
requestor	N_cst_appmanager instance to associate with this instance of n_cst_apppreference

Return value — Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage — The n_cst_appmanager of_SetAppPreference function calls this function.

Examples — This example is from the n_cst_appmanager of_SetAppPreference function:

```
...
IF ab_Switch THEN
  IF IsNull(inv_apppref) &
    OR NOT IsValid (inv_apppref) THEN
      inv_apppref = CREATE n_cst_apppreference
      inv_apppref. of_SetRequestor (this)
  Return 1
END IF
...
```

of_SetRestoreApp

Description — Specifies whether n_cst_apppreference restores application settings.

Access — Public

Syntax — *application.instancename*.**of_SetRestoreApp** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
boolean	Boolean specifying whether to save application preferences (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You typically call this function in the n_cst_appmanager Constructor event, just after enabling n_cst_apppreference.

Examples

This example calls the of_SetRestoreApp function:

```
...
IF this.of_IsRegistryAvailable() THEN
  this.inv_apppref.of_SetAppKey &
    (this.of_GetAppKey())
  this.inv_apppref.of_SetUserKey &
    (this.of_GetUserKey())
ELSE
  this.inv_apppref.of_SetAppINIFile &
    (this.of_GetAppINIFile())
  this.inv_apppref.of_SetUserINIFile &
    (this.of_GetUserINIFile())
END IF
this.inv_apppref.of_SetRestoreApp (TRUE)
this.inv_apppref.of_SetRestoreUser(TRUE)
```

of_SetRestoreUser

Description

Specifies whether n_cst_apppreference restores user settings.

Access

Public

Syntax

*application.instancename.***of_SetRestoreUser** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)

Argument	Description
boolean	Boolean specifying whether to save user preferences (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You typically call this function in the n_cst_appmanager Constructor event, just after enabling n_cst_apppreference.

Examples

This example calls the of_SetRestoreUser function:

```
...
this.of_SetAppPreference(TRUE)
IF this.of_IsRegistryAvailable() THEN
  this.inv_apppref.of_SetAppKey &
    (this.of_GetAppKey())
  this.inv_apppref.of_SetUserKey &
    (this.of_GetUserKey())
ELSE
  this.inv_apppref.of_SetAppINIFile &
    (this.of_GetAppINIFile())
  this.inv_apppref.of_SetUserINIFile &
    (this.of_GetUserINIFile())
END IF
this.inv_apppref.of_SetRestoreApp(TRUE)
this.inv_apppref.of_SetRestoreUser( TRUE)
```

of_SetUserIniFile

Description

Specifies the name of the INI file for user information. The application preference service uses this file by default on 16-bit Windows, UNIX, and the Macintosh.

Access

Public

Syntax

application.instancename.**of_SetUserIniFile** (*inifilename*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
inifilename	String specifying the fully qualified name of the user INI file

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The INI file for user information is typically the same as the n_cst_appmanager user INI file.

Examples

This example calls the of_SetUserIniFile function:

```
...
this.of_SetAppPreference(TRUE)
IF this.of_IsRegistryAvailable() THEN
  this.inv_apppref.of_SetAppKey &
    (this.of_GetAppKey())
  this.inv_apppref.of_SetUserKey &
    (this.of_GetUserKey())
ELSE
  this.inv_apppref.of_SetAppINIFile &
    (this.of_GetAppINIFile())
  this.inv_apppref.of_SetUserINIFile &
    (this.of_GetUserINIFile())
END IF
...
```

of_SetUserKey

Description

Specifies the name of the registry key for user information. N_cst_apppreference uses this key by default on Windows 95 and Windows NT.

Access

Public

Syntax

application.instancename.**of_SetUserKey** (*registrykey*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
registrykey	String containing the registry key

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The registry key for user information is typically the same as the n_cst_appmanager user key.

Examples

This example calls the of_SetUserKey function:

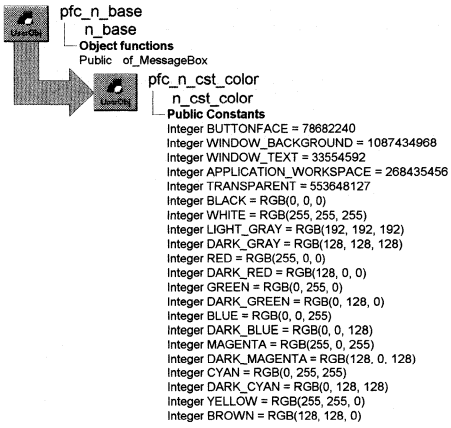
```
...
this.of_SetAppPreference(TRUE)
IF this.of_IsRegistryAvailable() THEN
  this.inv_apppref.of_SetAppKey &
    (this.of_GetAppKey())
  this.inv_apppref.of_SetUserKey &
    (this.of_GetUserKey())
ELSE
  this.inv_apppref.of_SetAppINIFile &
    (this.of_GetAppINIFile())
  this.inv_apppref.of_SetUserINIFile &
    (this.of_GetUserINIFile())
END IF
...
```

n_cst_color

Description

Color object. This object contains a set of constants you can reference to specify color values.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use this object:

1 Declare a variable of type n_cst_color.

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements:

```
n_cst_color lnv_color
```

2 Use n_cst_color constants as needed:

```
rte_1.SetTextColor  &
   (lnv_color.CYAN)
```

Instance variables

N_cst_color includes instance variables:

Instance variable	Description	Data type	Access	Usage
APPLICATION_WORKSPACE	Constant set to 268435456	Integer	Public	Specify colors

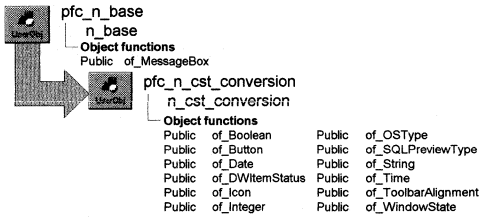
Instance variable	Description	Data type	Access	Usage
BLACK	Constant set to RGB(0, 0, 0)	Integer	Public	Specify colors
BLUE	Constant set to RGB(0, 0, 255)	Integer	Public	Specify colors
BROWN	Constant set to RGB(128, 128, 0)	Integer	Public	Specify colors
BUTTONFACE	Constant set to 78682240	Integer	Public	Specify colors
CYAN	Constant set to RGB(0, 255, 255)	Integer	Public	Specify colors
DARK_BLUE	Constant set to RGB(0, 0, 128)	Integer	Public	Specify colors
DARK_CYAN	Constant set to RGB(0, 128, 128)	Integer	Public	Specify colors
DARK_GRAY	Constant set to RGB(128, 128, 128)	Integer	Public	Specify colors
DARK_GREEN	Constant set to RGB(0, 128, 0)	Integer	Public	Specify colors
DARK_MAGENTA	Constant set to RGB(128, 0, 128)	Integer	Public	Specify colors
DARK_RED	Constant set to RGB(128, 0, 0)	Integer	Public	Specify colors
GREEN	Constant set to RGB(0, 255, 0)	Integer	Public	Specify colors
LIGHT_GRAY	Constant set to RGB(192, 192, 192)	Integer	Public	Specify colors
MAGENTA	Constant set to RGB(255, 0, 255)	Integer	Public	Specify colors
RED	Constant set to RGB(255, 0, 0)	Integer	Public	Specify colors
TRANSPARENT	Constant set to 553648127	Integer	Public	Specify colors
WHITE	Constant set to RGB(255, 255, 255)	Integer	Public	Specify colors
WINDOW_BACKGROUND	Constant set to 1087434968	Integer	Public	Specify colors
WINDOW_TEXT	Constant set to 33554592	Integer	Public	Specify colors
YELLOW	Constant set to RGB(255, 255, 0)	Integer	Public	Specify colors

n_cst_conversion

Description

Data type conversion service. This service provides functions you can call to convert values from one data type to another. For example, you can call the of_Boolean function to convert an integer or a string into a boolean value.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use conversion functions:

1 Declare a variable of type n_cst_conversion.

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements:

```
n_cst_conversion   inv_conversion
```

2 Call conversion service functions as needed.

This example assumes an inv_conversion instance variable:

```
String    ls_fulltext
Boolean   lb_fulltext

ls_fulltext = ProfileString &
   (gnv_app.of_GetUserIniFile( ), &
      "Display", "FullText", "Yes")
lb_fulltext = &
   inv_conversion.of_Boolean( ls_fulltext )
```

See also

n_cst_numerical
n_cst_string

Functions

N_cst_conversion includes precoded object functions:

of_Boolean	of_OSType
of_Button	of_SQLPreviewType
of_Date	of_String
of_DWItemStatus	of_Time
of_Icon	of_ToolbarAlignment
of_Integer	of_WindowState

of_Boolean

Description

Returns a boolean representation of the passed integer or string.

Access

Public

Syntax

instancename.**of_Boolean** (*stringorint*)

Argument	Description
instancename	Instance name of n_cst_conversion
stringorint	String or integer to be converted to a boolean value

Return value

Boolean. Returns values as follows:

- **If *stringorint* is an integer** Returns TRUE if *stringorint* is 1, FALSE if *stringorint* is 0, and NULL if *stringorint* is any other value or NULL.
- **If *stringorint* is a string** Returns TRUE if *stringorint* is true, t, yes, y, or 1. Returns FALSE if *stringorint* is false, f, no, n or 0. This function returns NULL if *stringorint* is any other value or NULL.

Usage

Call this function to convert a string or integer to a boolean value.

Examples

This example calls the of_Boolean function to convert an INI file value to a boolean. It assumes an inv_conversion instance variable:

```
String    ls_fulltext
Boolean   lb_fulltext

ls_fulltext = ProfileString &
   (gnv_app.of_GetUserIniFile(), &
      "Display", "FullText", "Yes")
lb_fulltext = &
   inv_conversion.of_Boolean(ls_fulltext)
```

of_Button

Description

Converts a string value to a Button enumerated data type.

Access

Public

Syntax

instancename.**of_Button** (*string*, *button*)

Argument	Description
instancename	Instance name of n_cst_conversion
string	String or integer to be converted to a button. *String* must be one of the following: OK OKCancel YesNo YesNoCancel RetryCancel AbortRetryIgnore
button	Button enumerated data type to contain the converted string (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The error service calls this function to convert error message properties into the buttons displayed on message boxes.

Examples

This example calls the of_Button function:

```
...
IF ll_row > 0 THEN
   inv_conversion.of_Button &
      (ids_messages.object.button[ll_row], &
      inv_errorpass.ie_buttonstyle)
   inv_conversion.of_Icon &
      (ids_messages.object.icon[ll_row], &
      inv_errorpass.ie_icon)
...
```

of_Date

Description

Converts a string value in date or datetime format to a date data type.

Access

Public

Syntax

instancename.**of_Date** (*string*)

Argument	Description
instancename	Instance name of n_cst_conversion
string	String in date or datetime format to be converted to a date data type

Return value

Date. Returns a date if the function succeeds, 1900-01-01if *string* is an invalid date or datetime, and NULL if *string* is NULL.

Usage

Compare this function with the PowerScript function, which requires a string containing a date value as the argument.

Examples

This example calls the of_Date function:

```
String    ls_datetimeval = "07/28/71 12:35:14"
Date      ld_date
Time      ltm_time
n_cst_conversion   lnv_conversion

ld_date = lnv_conversion.of_Date(ls_datetimeval)
ltm_time = lnv_conversion.of_Time(ls_datetimeval)
...
```

of_DWItemStatus

Description

Converts a string value to a dwItemStatus enumerated data type.

Access

Public

Syntax

instancename.**of_DWItemStatus** (*originalvalue*, *dwitemstatus*)

Argument	Description
instancename	Instance name of n_cst_conversion
originalvalue	String to be converted to the dwItemStatus enumerated data type. Valid values are: DataModified New NewModified NotModified
dwitemstatus	DWItemStatus enumerated data type to contain the converted string (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, -1 if *originalvalue* is not a valid value, and NULL if *originalvalue* is NULL.

Usage

The DataWindow Properties dialog box calls this function to convert a string to a DWItemStatus enumerated data type.

Examples

This example calls the of_DWItemStatus function:

```
Integer   li_rc
Long   ll_currow
String   ls_desiredstatus
dwItemStatus le_desiredstatus
n_cst_conversion lnv_conversion

ll_currow = dw_requestorview.GetRow()
IF ll_currow > 0 THEN
   ls_desiredstatus = "NotModified"
   li_rc = lnv_conversion.of_DWItemStatus  &
      (ls_desiredstatus, le_desiredstatus)
...
```

of_Icon

Description

Converts a string value to an Icon enumerated data type.

Access

Public

Syntax

instancename.**of_Icon** (*string, icon*)

Argument	Description
instancename	Instance name of n_cst_conversion
string	String or integer to be converted to an icon. *String* must be one of the following: None Question Information StopSign Exclamation
icon	Icon enumerated data type to contain the converted string (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The error service calls this function to convert error message properties into the icons displayed on message boxes.

Examples

This example calls the of_Icon function:

```
...
IF ll_row > 0 THEN
   inv_conversion.of_Button &
      (ids_messages.object.button[ll_row], &
      inv_errorpass.ie_buttonstyle)
   inv_conversion.of_Icon &
      (ids_messages.object.icon[ll_row], &
      inv_errorpass.ie_icon)
...
```

of_Integer

Description

Converts a boolean argument to an integer value (converts TRUE to 1 and FALSE to 0).

Access

Public

Syntax

instancename.**of_Integer** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_conversion
boolean	Boolean to be converted to an integer

Return value

Integer. Returns 1 if *boolean* is TRUE and 0 if *boolean* is FALSE.

Examples

This example calls of_Integer to convert a boolean value to an integer (assuming an inv_conversion instance variable):

```
Integer   li_boolean

li_boolean = inv_conversion.of_Integer &
   (cbx_minimize.Checked)
```

of_OSType

Description

Converts a string value to an OSTypes enumerated data type.

Access

Public

Syntax

instancename.**of_OSType** (*originalvalue*, *ostype*)

Argument	Description
instancename	Instance name of n_cst_conversion
originalvalue	String to be converted to the OSType enumerated data type. Valid values are: AIX HPUX Macintosh OSF1 SOL2 Windows WindowsNT
ostype	OSTypes variable into which the function returns the value (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, -1 if *originalvalue* is not a valid value, and NULL if *originalvalue* is NULL.

Examples

This example calls the of_OSType function:

```
OSTypes  lost_type
String   ls_original
Integer  li_return
n_cst_conversion  lnv_conversion

ls_original = "Windows"
li_return = lnv_conversion.of_OSType  &
 (ls_original, lost_type)
...
```

of_SQLPreviewType

Description

Converts a string value to the SQLPreviewType enumerated data type.

Access

Public

Syntax

instancename.**of_SQLPreviewType** (*originalvalue*, *sqlpreview*)

Argument	Description
instancename	Instance name of n_cst_conversion

Argument	Description
originalvalue	String to be converted to the SQLPreviewType enumerated data type. Valid values are: Insert Update Delete Retrieve
sqlpreview	SQLPreviewType variable into which the function returns the value (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, -1 if *originalvalue* is not a valid value, and NULL if *originalvalue* is NULL.

Examples

This example calls the of_SQLPreviewType function:

```
SQLPreviewType   lsql_type
String    ls_statement

ls_statement = "Insert"
inv_conversion.of_SQLPreviewType &
   (ls_statement, lsql_type)
```

of_String

Converts various data types to a string. There are two syntaxes, depending on the original data type:

To	Use
Convert a boolean to a string	Syntax 1
Convert SQLPreviewType, ToolbarAlignment, Icon, Button, DWItemStatus, WindowState, or OSTypes to a string	Syntax 2

Syntax 1

To convert a boolean to a string

Description

Creates a string from a boolean.

Access

Public

Syntax

instancename.**of_String** (*boolean* {, *returntype* })

Argument	Description
instancename	Instance name of n_cst_conversion

Argument	Description
boolean	Boolean value to be converted to a string
returntype (optional)	String specifying how the function formats the return value. Valid values are: ♦ TRUEFALSE Returns TRUE or FALSE (default) ♦ TF Returns T or F ♦ YESNO Returns YES or NO ♦ YN Returns Y or N ♦ ZEROONE Returns 1 or 0

Return value

String. Returns a string corresponding to the passed boolean value. The string's contents vary depending on the *returntype* argument.

Usage

Use Syntax 2 to convert a SQLPreviewType, ToolbarAlignment, Icon, Button, DWItemStatus, WindowState, or OSTypes to a string.

Examples

This example calls the of_String function:

```
Date      ldt_date
Boolean   lb_leapyear

ldt_date = Today( )
lb_leapyear = inv_datetime.of_IsLeapYear(ldt_date)
MessageBox("Leap Year Test", &
    "Leap year = " + &
       inv_conversion.of_String(lb_leapyear,'YESNO'))
```

Syntax 2

To convert a SQLPreviewType, ToolbarAlignment, Icon, Button, DWItemStatus, WindowState, or OSTypes

Description

Creates a string from a SQLPreviewType, ToolbarAlignment, Icon, Button, DWItemStatus, WindowState, or OSTypes enumerated variable.

Access

Public

Syntax

instancename.**of_String** (*input*)

Argument	Description
instancename	Instance name of n_cst_conversion
input	SQLPreviewType, ToolbarAlignment, Icon, Button, DWItemStatus, WindowState, or OSTypes enumerated variable to be converted to a string

Return value

String. Returns values as follows:

- **If *input* is a SQLPreviewType** Returns one of the following values:

  Delete
  Insert
  Retrieve
  Update
  ! (invalid argument)
  Returns one of the following values:

- **If *input* is a ToolbarAlignment** Returns one of the following values:

  Bottom
  Floating
  Left
  Right
  Top
  ! (invalid argument)

- **If *input* is an Icon** Returns one of the following values:

  None
  Information
  StopSign
  Information
  Exclamation
  ! (invalid argument)

- **If *input* is a Button** Returns one of the following values:

  OK
  OKCancel
  YesNo
  YesNoCancel
  RetryCancel
  AbortRetryIgnore
  ! (invalid argument)

- **If *input* is a DWItemStatus** Returns one of the following values:

  DataModified
  New
  NewModified
  NotModified
  ! (invalid argument)

- **If *input* is a WindowState** Returns one of the following values:

  Maximized
  Minimized

Normal
! (invalid argument)

- **If *input* is an OSTypes** Returns one of the following values:

  AIX
  HPUX
  Macintosh
  OSF1
  SOL2
  Windows
  WindowsNT
  ! (invalid argument)

Usage

Call this function to convert a SQLPreviewType, ToolbarAlignment, Icon, Button, DWItemStatus, WindowState, or OSTypes enumerated variable to a string.

Use Syntax 1 to convert a boolean to a string.

Examples

This example calls the of_String function to convert a ToolbarAlignment to a string:

```
String    ls_string

ls_string = inv_conversion.of_String &
   (gnv_app.of_GetFrame().ToolbarAlignment)
```

of_Time

Description

Converts a string value in time or datetime format to a time data type.

Access

Public

Syntax

instancename.**of_Time** (*string*)

Argument	Description
instancename	Instance name of n_cst_conversion
string	String in time or datetime format to be converted to a time data type

Return value

Time. Returns a time if the function succeeds, 00:00:00.000000 if *string* is an invalid time or datetime, and NULL if *string* is NULL.

Examples

This example calls the of_Time function:

```
String    ls_datetimeval = "07/28/71 12:35:14"
Date      ld_date
Time      ltm_time
n_cst_conversion   lnv_conversion

ld_date = lnv_conversion.of_Date(ls_datetimeval)
ltm_time = lnv_conversion.of_Time(ls_datetimeval)
...
```

of_ToolbarAlignment

Description

Converts a string value to the ToolbarAlignment enumerated data type.

Access

Public

Syntax

instancename.**of_ToolbarAlignment** (*originalvalue*, *toolbaralign*)

Argument	Description
instancename	Instance name of n_cst_conversion
originalvalue	String to be converted to the SQLPreviewType enumerated data type. Valid values are: Bottom Floating Left Right Top
toolbaralign	ToolbarAlignment variable into which the function returns the value (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, -1 if *originalvalue* is not a valid value, and NULL if *originalvalue* is NULL.

Examples

This example calls the of_ToolbarAlignment function:

```
ToolbarAlignment   le_alignment
Integer            li_index, li_rc
n_cst_conversion   lnv_conversion

li_index = of_GetListBoxIndex &
   (lb_toolbar.SelectedItem())
IF li_index > 0 THEN
   li_rc = lnv_conversion.of_ToolbarAlignment &
      (This.text, le_alignment)
```

```
    IF li_rc = 1 THEN
       istr_toolbar[li_index].e_alignment = &
          le_alignment
    END IF
END IF
```

of_WindowState

Description

Converts a string value to a WindowState enumerated data type.

Access

Public

Syntax

instancename.**of_WindowState** (*string*, *windowstate*)

Argument	Description
instancename	Instance name of n_cst_conversion
string	String or integer to be converted to a windowstate. *String* must be one of the following: Maximized Minimized Normal
windowstate	WindowState enumerated data type to contain the converted string (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the n_cst_error of_GetError function:

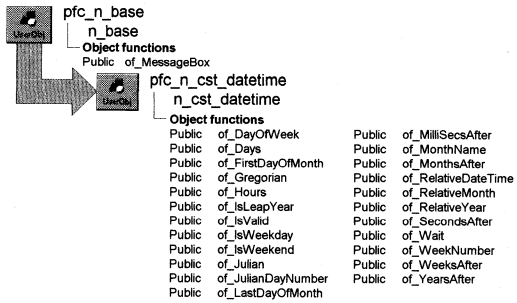
```
WindowState   lws_state
Integer       li_return
n_cst_conversion lnv_conversion

li_return = lnv_conversion.of_WindowState &
   ("maximized", lws_state)
```

n_cst_datetime

Description

Date/time service. This service provides functions that you can call to perform calculations with dates. For example, you can call the of_SecondsAfter function to determine the number of seconds between two date/time values.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use date/time functions:

1 Declare a variable of type n_cst_datetime.

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements:

```
n_cst_datetime   inv_datetime
```

2 Call date/time service functions as needed.

This example assumes an inv_datetime instance variable:

```
Long   ll_seconds, ll_days

ll_seconds = Long(sle_seconds.Text)
ll_days = inv_datetime.of_Days(ll_seconds)

MessageBox("Date/Time", String(ll_seconds) &
   + " seconds is equal to " + &
      String(ll_days) + " days.")
```

See also

n_cst_conversion
n_cst_filesrv
n_cst_numerical
n_cst_platform
n_cst_sql

n_cst_string

Functions

N_cst_datetime includes precoded object functions:

of_DayOfWeek	of_MilliSecsAfter
of_Days	of_MonthName
of_FirstDayOfMonth	of_MonthsAfter
of_Gregorian	of_RelativeDateTime
of_Hours	of_RelativeMonth
of_IsLeapYear	of_RelativeYear
of_IsValid	of_SecondsAfter
of_IsWeekDay	of_Wait
of_IsWeekEnd	of_WeekNumber
of_Julian	of_WeeksAfter
of_JulianDayNumber	of_YearsAfter
of_LastDayOfMonth	

of_DayOfWeek

Description

Determines the day of the week for the passed date.

Access

Public

Syntax

instancename.**of_DayOfWeek** (*sourcedate*)

Argument	Description
instancename	Instance name of n_cst_datetime
sourcedate	Date variable that contains the day to be determined

Return value

Integer. Returns values, as follows:

1 Sunday
2 Monday
3 Tuesday
4 Wednesday
5 Thursday
6 Friday
7 Saturday

If *sourcedate* is invalid, the function returns -1; if *sourcedate* is NULL, the function returns NULL.

Usage

Call this function to convert a date variable into a weekday.

Examples

This example calls the of_DayOfWeek function:

```
Date      ldt_date
Integer   li_day
String    ls_days[ ] = &
   {"Sunday", "Monday", "Tuesday", "Wednesday", &
      "Thursday", "Friday", "Saturday"}

ldt_date = Today()
li_day = inv_datetime.of_DayOfWeek(ldt_date)
MessageBox("Convert Date to Day", &
   "Today is " + ls_days[li_day])
```

of_Days

Description

Given the number of seconds, calculates the equivalent number of days.

Access

Public

Syntax

instancename.**of_Days** (*seconds*)

Argument	Description
instancename	Instance name of n_cst_datetime
seconds	Long specifying the number of seconds to be converted to days

Return value

Long. Returns the number of days for *seconds*. If *seconds* is invalid, the function returns -1; if *seconds* is NULL, the function returns NULL.

Examples

This example calls the of_Days function:

```
Long   ll_seconds, ll_days

ll_seconds = Long(sle_seconds.Text)
ll_days = inv_datetime.of_Days(ll_seconds)
MessageBox("Date/Time", &
   String(ll_seconds) + " seconds is equal to " + &
      String(ll_days) + " days.")
```

of_FirstDayOfMonth

Description

Given a date, determines the first date of the month.

Access

Public

Syntax

instancename.**of_FirstDayOfMonth** (*sourcedate*)

Argument	Description
instancename	Instance name of n_cst_datetime
sourcedate	Date for which the date of the first day of the month is returned

Return value

Date. Returns the date of the first day of the month. If *sourcedate* is invalid, the function returns 1900-01-01; if *sourcedate* is NULL, the function returns NULL.

Examples

This example calls the of_FirstDayOfMonth function:

```
Date   ldt_date, ldt_firstday

ldt_date = Today( )
ldt_firstday = &
   inv_datetime.of_FirstDayOfMonth(ldt_date)
MessageBox("First day of the month", &
   "Today is " + String(ldt_date) + &
      " - First day of the month is " + &
         String(ldt_firstday))
```

of_Gregorian

Description

Converts a Julian date to a Gregorian date.

Access

Public

Syntax

instancename.**of_Gregorian** (*juliandate*)

Argument	Description
instancename	Instance name of n_cst_datetime
juliandate	Long containing the Julian date to be converted to Gregorian (the basis for *juliandate* is January 1, 1900)

Return value

Date. Returns the Gregorian date that corresponds to *juliandate*. If *juliandate* is NULL, the function returns NULL.

Examples

This example calls the of_Gregorian function:

```
Long   ll_julian
Date   ldt_gregorian
```

```
ll_julian = Long(sle_julian.Text)
ldt_gregorian = inv_datetime.of_Gregorian &
   (ll_julian)
   MessageBox("Julian to Gregorian", &
      "Julian: " + String(ll_julian) + &
         " is Gregorian: " + String(ldt_gregorian))
```

of_Hours

Description

Converts seconds to hours.

Access

Public

Syntax

instancename.**of_Hours** (*seconds*)

Argument	Description
instancename	Instance name of n_cst_datetime
seconds	Long containing the seconds value to be converted

Return value

Long. Returns the number of hours for *seconds*. If *seconds* is invalid, the function returns -1; if *seconds* is NULL, the function returns NULL.

Examples

This example calls the of_Hours function:

```
Long   ll_seconds, ll_hours

ll_seconds = Long(sle_seconds.Text)
ll_hours = inv_datetime.of_Hours(ll_seconds)
MessageBox("Date/Time", &
   String(ll_seconds) + " seconds is equal to " + &
      String(ll_hours) + " hours.")
```

of_IsLeapYear

Description

Determines whether the passed date is in a leap year.

Access

Public

Syntax

instancename.**of_IsLeapYear** (*sourcedate*)

Argument	Description
instancename	Instance name of n_cst_datetime

Argument	Description
sourcedate	Date variable containing the date to be tested

Return value

Boolean. Returns TRUE if *sourcedate* is in a leap year, FALSE if it is not, and NULL if it is NULL.

Examples

This example calls the of_IsLeapYear function:

```
Date      ldt_date
Boolean   lb_leapyear

ldt_date = Today( )
lb_leapyear = inv_datetime.of_IsLeapYear(ldt_date)
MessageBox("Leap Year Test", &
    "Leap year = " + &
       inv_conversion.of_String(lb_leapyear,'YESNO'))
```

of_IsValid

Description

Reports whether a passed argument is a valid date, DateTime, or time.

Access

Public

Syntax

instancename.**of_IsValid** (*source*)

Argument	Description
instancename	Instance name of n_cst_datetime
source	Date, DateTime, or Time value to be tested

Return value

Boolean. Returns TRUE if *source* is a valid date, DateTime, or time; FALSE if it is not; and NULL is *source* is NULL.

Usage

Call this function to ensure valid data before updating the database or calling functions.

Examples

This example calls the of_IsValid function:

```
Boolean   lb_valid

lb_valid = &
   inv_datetime.of_IsValid(Date(sle_date.Text))
IF NOT lb_valid THEN
   MessageBox("Date/Time", &
```

```
        sle_date.text + " is not a valid date")
END IF
```

of_IsWeekDay

Description

Determines if a passed date is a weekday.

Access

Public

Syntax

instancename.**of_IsWeekDay** (*source*)

Argument	Description
instancename	Instance name of n_cst_datetime
source	Date variable to be tested

Return value

Boolean. Returns TRUE if *source* is a weekday, FALSE if it is not, and NULL if *source* is NULL.

Examples

This example calls the of_IsWeekDay function:

```
Date      ldt_date
Boolean   lb_weekday

ldt_date = Today()
lb_weekday = inv_datetime.of_IsWeekDay(ldt_date)
IF lb_weekday THEN
   MessageBox("Weekday Test", String(ldt_date) &
      + " is a weekday")
ELSE
   MessageBox("Weekday Test", String(ldt_date) &
      + " is not a weekday")
END IF
```

of_IsWeekEnd

Description

Determines if a passed date is on a weekend.

Access

Public

Syntax

instancename.**of_IsWeekEnd** (*source*)

Argument	Description
instancename	Instance name of n_cst_datetime

Argument	Description
source	Date variable to be tested

Return value

Boolean. Returns TRUE if *source* is on a weekend, FALSE if it is not, and NULL if *source* is NULL.

Examples

This example calls the of_IsWeekEnd function:

```
Date      ldt_date
Boolean   lb_weekend

ldt_date = Today( )
lb_weekend = inv_datetime.of_IsWeekEnd(ldt_date)
IF lb_weekend THEN
   MessageBox("Weekend Test", String(ldt_date) &
      + " is a weekend")
ELSE
   MessageBox("Weekend Test", String(ldt_date) &
      + " is not a weekend")
END IF
```

of_Julian

Description

Converts a Gregorian date to a Julian date.

Access

Public

Syntax

instancename.**of_Julian** (*gregoriandate*)

Argument	Description
instancename	Instance name of n_cst_datetime
gregoriandate	Date containing the Gregorian date to be converted to Julian (the basis for *gregoriandate* is January 1, 1900)

Return value

Long. Returns the Julian date that corresponds to *gregoriandate*. If *gregoriandate* is invalid, the function returns -1; if it is NULL, the function returns NULL.

Examples

This example calls the of_Julian function:

```
Date   ldt_gregorian
Long   ll_julian
```

```
ldt_gregorian = Today( )
ll_julian = inv_datetime.of_Julian &
   (ldt_gregorian)
   MessageBox("Gregorian to Julian", &
      "Gregorian: " + String(ldt_gregorian) + &
         " is Julian: " + String(ll_julian))
```

of_JulianDayNumber

Description

Given a date, this function determines the day number, within the current year.

Access

Public

Syntax

instancename.**of_JulianDayNumber** (*source*)

Argument	Description
instancename	Instance name of n_cst_datetime
source	Date for which day number is returned

Return value

Long. Returns the day number within the current year.

Examples

This example calls the of_JulianDayNumber function:

```
Date   ldt_gregorian
Long   ll_julian

ldt_gregorian = Today( )
ll_julian = inv_datetime.of_JulianDayNumber &
   (ldt_gregorian)
   MessageBox("Gregorian to Julian", &
      "Gregorian: " + String(ldt_gregorian) + &
         " is day number: " + String(ll_julian))
```

of_LastDayOfMonth

Description

Given a date, this function determines the last date of the date's month.

Access

Public

Syntax

instancename.**of_LastDayOfMonth** (*sourcedate*)

Argument	Description
instancename	Instance name of n_cst_datetime

Argument	Description
sourcedate	Date for which the date of the last day of the month is returned

Return value

Date. Returns the date of the last day of the month. If *sourcedate* is invalid, the function returns 1900-01-01; if *sourcedate* is NULL, the function returns NULL.

Examples

This example calls the of_LastDayOfMonth function:

```
Date   ldt_date, ldt_lastday

ldt_date = Today( )
ldt_lastday = &
   inv_datetime.of_LastDayOfMonth(ldt_date)
MessageBox("Last day of the month", &
   "Today is " + String(ldt_date) + &
      " - Last day of the month is " + &
         String(ldt_lastday) )
```

of_MilliSecsAfter

Description

Calculates the number of milliseconds between two times.

Access

Public

Syntax

instancename.**of_MilliSecsAfter** (*starttime*, *endtime*)

Argument	Description
instancename	Instance name of n_cst_datetime
starttime	Time variable containing the starting time
endtime	Time variable containing the ending time

Return value

Long. Returns the number of milliseconds between *starttime* and *endtime*. If *endtime* is less than *starttime*, this value is negative. If either argument is NULL, the function returns a NULL.

Examples

This example calls the of_MilliSecsAfter function:

```
Time   lt_start, lt_finish
Long   ll_between, ll_loop, ll_count

lt_start = Now( )
```

```
For ll_loop = 1 to 10000
   ll_count++
NEXT
lt_finish = Now( )
ll_between = inv_datetime.of_MilliSecsAfter &
   (lt_start, lt_finish)
   MessageBox("Date/Time", String(ll_between) &
      + " millisecs between " + String(lt_start) &
         + " and " + String(lt_finish) )
```

of_MonthName

Description

Given the month number or a date, calculates the month name.

Access

Public

Syntax

instancename.**of_MonthName** (*numberordate*)

Argument	Description
instancename	Instance name of n_cst_datetime
numberordate	Date variable or month number for which the function returns the month name

Return value

String. Returns the month name.

Examples

This example calls the of_MonthName function:

```
Date   ldt_date
String ls_name
n_cst_datetime lnv_datetime

ldt_date = Today()

ls_name = lnv_datetime.of_MonthName &
   (ldt_date)
MessageBox("Date/Time", ls_name)
```

of_MonthsAfter

Description

Calculates the number of whole months between two dates.

Access

Public

Syntax

instancename.**of_MonthsAfter** (*startdate, enddate*)

Argument	Description
instancename	Instance name of n_cst_datetime
startdate	Date variable containing the start date
enddate	Date variable containing the end date

Return value

Long. Returns the number of months between *startdate* and *enddate*. If *enddate* is less than *startdate*, this value is negative. If either argument is NULL or invalid, the function returns a NULL.

Examples

This example calls the of_MonthsAfter function:

```
Date   ldt_start, ldt_finish
Long   ll_between

ldt_start = Date(sle_startmonth.text)
ldt_finish = Date(sle_endmonth.text)

ll_between = inv_datetime.of_MonthsAfter &
   (ldt_start, ldt_finish)
MessageBox("Date/Time", String(ll_between) &
   + " months between " + sle_startmonth.text &
      + " and " + sle_endmonth.text )
```

of_RelativeDateTime

Description

Determines a date/time relative to a given DateTime.

Access

Public

Syntax

instancename.**of_RelativeDateTime** (*start, offset*)

Argument	Description
instancename	Instance name of n_cst_datetime
start	Datetime variable that the function uses as the starting point
offset	Long representing the number of seconds between *start* and the returned relative date/time (can be either positive or negative)

Return value

Datetime. Returns the date/time *offset* number of seconds before or after *start*. If *start* is invalid, the function returns 1900-01-01; if *start* is NULL, the function returns NULL.

Examples

This example calls the of_RelativeDateTime function:

```
Datetime   ldtm_start, ldtm_new
Long       ll_between

ldtm_start = DateTime(Today( ), Now( ) )
ll_between = Long(sle_relative.text)

ldtm_new = inv_datetime.of_RelativeDateTime &
   (ldtm_start, ll_between)
MessageBox("Date/Time", String(ldtm_new) &
   + " is " + sle_relative.text &
      + " seconds before/after " + string(ldtm_start))
```

of_RelativeMonth

Description

Calculates a date, plus or minus a given number of months from a given date. For example, three months from 6/13/97 is 9/13/97.

Access

Public

Syntax

instancename.**of_RelativeMonth** (*start*, *offset*)

Argument	Description
instancename	Instance name of n_cst_datetime
start	Date variable that the function uses as the starting point
offset	Long representing the number of months between *start* and the returned relative date (can be either positive or negative)

Return value

Date. Returns a date the specified number of months before or after *start*. If *start* is invalid, the function returns 1900-01-01; if *start* is NULL, the function returns NULL.

Examples

This example calls the of_RelativeMonth function:

```
Date   ldt_start, ldt_new
Long   ll_between

ldt_start = Today( )
ll_between = Long(sle_relative.text)

ldt_new = inv_datetime.of_RelativeMonth &
   (ldt_start, ll_between)
```

```
MessageBox("Date/Time", String(ldt_new) &
   + " is " + sle_relative.text &
      + " months before/after " + string(ldt_start )
```

of_RelativeYear

Description

Calculates a date, plus or minus a given number of years from a given date. For example, three years from 6/13/97 is 6/13/00.

Access

Public

Syntax

instancename.**of_RelativeYear** (*start, offset*)

Argument	Description
instancename	Instance name of n_cst_datetime
start	Date variable that the function uses as the starting point
offset	Long representing the number of years between *start* and the returned relative date (can be either positive or negative)

Return value

Date. Returns a date the specified number of years before or after *start*. If *start* is invalid, the function returns 1900-01-01; if *start* is NULL, the function returns NULL.

Examples

This example calls the of_RelativeYear function. It assumes an inv_datetime instance variable:

```
Date   ldt_start, ldt_new
Long   ll_between

ldt_start = Today( )
ll_between = Long(sle_relative.text)

ldt_new = inv_datetime.of_RelativeYear &
   (ldt_start, ll_between)
MessageBox("Date/Time", String(ldt_new) &
   + " is " + sle_relative.text &
      + " years before/after " + string(ldt_start))
```

of_SecondsAfter

Description

Calculates the number of seconds between two date/times.

Access

Public

Syntax

instancename.**of_SecondsAfter** (*starttime*, *endtime*)

Argument	Description
instancename	Instance name of n_cst_datetime
starttime	Datetime variable containing the starting time
endtime	Datetime variable containing the ending time

Return value

Long. Returns the number of seconds between *starttime* and *endtime*. If *endtime* is less than *starttime*, this value is negative. If either argument is NULL or invalid, the function returns a NULL.

Examples

This example calls the of_SecondsAfter function:

```
Datetime   ldtm_start, ldtm_finish
Long       ll_between, ll_loop, ll_count

ldtm_start = DateTime(Today( ), Now( ) )
For ll_loop = 1 to 100000
   ll_count++
NEXT
ldtm_finish = DateTime(Today( ), Now( ) )
ll_between = inv_datetime.of_SecondsAfter &
   (ldtm_start, ldtm_finish)
   MessageBox("Date/Time", String(ll_between) &
      + " seconds between " + String(ldtm_start) &
         + " and " + String(ldtm_finish) )
```

of_Wait

Stops processing temporarily. There are two syntaxes:

To	Use
Wait until a specified date/time	Syntax 1
Wait a specified number of seconds	Syntax 2

Syntax 1

Wait until a specified date/time

Description

Stops processing until a specified date/time is reached.

Access

Public

Syntax

instancename.**of_Wait** (*waituntil*)

Argument	Description
instancename	Instance name of n_cst_datetime
waituntil	Datetime variable that must be reached before processing will continue

Return value

Integer. Returns 1 when the wait finishes, -1 if *waituntil* is invalid, or NULL if *waituntil* is NULL.

Usage

Use Syntax 2 to wait a specified number of seconds.

Examples

This example calls the of_Wait function. It assumes an inv_datetime instance variable:

```
Datetime   ldtm_start, ldtm_wait
Long       ll_wait
Integer    li_return

ldtm_start = DateTime(Today( ), Now( ) )
ll_wait = Long(sle_wait.text)
ldtm_wait = inv_datetime.of_RelativeDateTime &
   (ldtm_start, ll_wait)
gnv_app.of_GetFrame( ).SetMicroHelp &
   ("Begin wait: " + String(Now()))
li_return = inv_datetime.of_Wait(ldtm_wait)
gnv_app.of_GetFrame().SetMicroHelp("End wait: " &
   + String(Now ()))
```

Syntax 2

Wait a specified number of seconds

Description

Stops processing for a specified number of seconds.

Access

Public

Syntax

instancename.**of_Wait** (*waitseconds*)

Argument	Description
instancename	Instance name of n_cst_datetime
waitseconds	UnsignedLong specifying the number of seconds to wait

Return value

Integer. Returns 1 when the wait finishes, -1 if *waituntil* is invalid, and NULL if *waituntil* is NULL.

Usage

Use Syntax 1 to wait until a specified date/time.

Examples

This example calls the of_Wait function. It assumes an inv_datetime instance variable:

```
Datetime   ldtm_start, ldtm_wait
Long       ll_wait
Integer    li_return

ll_wait = Long(sle_wait.text)
gnv_app.of_GetFrame( ).SetMicroHelp("Begin wait: " &
   + String(Now()))
li_return = inv_datetime.of_Wait(ll_wait)
gnv_app.of_GetFrame().SetMicroHelp("End wait: " &
   + String(Now ()))
```

of_WeekNumber

Description

Calculates the week number for a specified date (within the date's year).

Access

Public

Syntax

instancename.**of_WeekNumber** (*date*)

Argument	Description
instancename	Instance name of n_cst_datetime
date	Date variable containing the date for which the week number is returned

Return value

Long. Returns the week number that corresponds to the passed date. If *date* is invalid, the function returns -1; if *date* is NULL, the function returns NULL.

Examples

This example calls the of_WeekNumber function. It assumes an inv_datetime instance variable:

```
Date   ldt_date
Long   ll_weeknumber

ldt_date = Today()
ll_weeknumber = inv_datetime.of_WeekNumber(ldt_date)
MessageBox("Week Number", String(ldt_date) &
   + " is week number " + String(ll_weeknumber) )
```

of_WeeksAfter

Description

Calculates the number of whole weeks between two dates.

Access

Public

Syntax

instancename.**of_WeeksAfter** (*startdate, enddate*)

Argument	Description
instancename	Instance name of n_cst_datetime
startdate	Date variable containing the start date
enddate	Date variable containing the end date

Return value

Long. Returns the number of weeks between *startdate* and *enddate*. If *enddate* is less than *startdate*, this value is negative; if either argument is NULL or invalid, the function returns a NULL.

Examples

This example calls the of_WeeksAfter function. It assumes an inv_datetime instance variable:

```
Date   ldt_start, ldt_finish
Long   ll_between

ldt_start = Date(sle_startweek.text)
ldt_finish = Date(sle_endweek.text)

ll_between = inv_datetime.of_WeeksAfter &
(ldt_start, ldt_finish)
MessageBox("Date/Time", String(ll_between) &
    + " weeks between " + sle_startweek.text &
       + " and " + sle_endweek.text )
```

of_YearsAfter

Description

Calculates the number of whole years between two dates.

Access

Public

Syntax

instancename.**of_YearsAfter** (*startdate, enddate*)

Argument	Description
instancename	Instance name of n_cst_datetime
startdate	Date variable containing the start date

Argument	Description
enddate	Date variable containing the end date

Return value

Long. Returns the number of years between *startdate* and *enddate*. If *enddate* is less than *startdate*, this value is negative; if either argument is NULL or invalid, the function returns a NULL.

Examples

This example calls the of_YearsAfter function. It assumes an inv_datetime instance variable:

```
Date   ldt_start, ldt_finish
Long   ll_between

ldt_start = Date(sle_startdate.text)
ldt_finish = Date(sle_enddate.text)
ll_between = inv_datetime.of_YearsAfter &
   (ldt_start, ldt_finish)
MessageBox("Date/Time", String(ll_between) &
   + " years between " + sle_startdate.text &
      + " and " + sle_enddate.text )
```

n_cst_debug

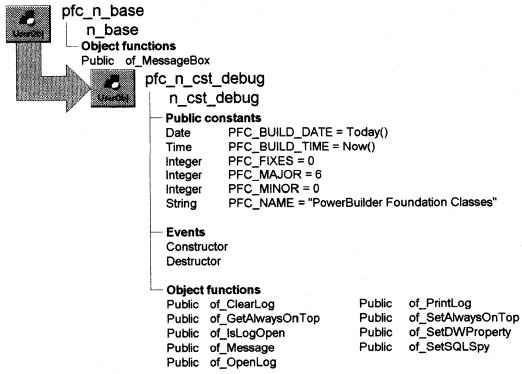
Description

Debugging object. This object includes:

- The reference variable for the SQL SPY debugging service
- A function to enable or disable the DataWindow Properties window
- Log window capabilities

When this object is instantiated, PFC displays debugging messages in certain situations.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_sqlspy
n_ds
u_dw
w_debuglog

Usage

To use this object:

1 Enable the debugging service by calling the n_cst_appmanager of_SetDebug function:

```
gnv_app.of_SetDebug(TRUE)
```

Once enabled, the service captures all entries to the log, even if you don't open the w_debuglog window.

2 (Optional) Enable the SQL Spy service by calling the n_cst_debug of_SetSQLSpy function:

```
gnv_app.inv_debug.of_SetSQLSpy(TRUE)
```

3 (Optional) Log debugging messages:

```
gnv_app.inv_debug.of_Message &
   ("Opening window w_test")
```

4 (Optional) Display logged messages:

```
gnv_app.inv_debug.of_OpenLog(TRUE)
```

5 (Optional) Enable the DataWindow Properties window by calling the n_cst_debug of_SetDWProperty function:

```
gnv_app.inv_debug.of_SetDWProperty(TRUE)
```

See also

n_cst_appmanager
n_cst_error

Instance variables

N_cst_debug includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_alwaysontop	Specifies whether the service displays the debug window on top of other windows	Boolean	Protected	Call of_SetAlwaysOnTop and of_GetAlwaysOnTop to access
PFC_BUILD_DATE	Build date	Date	Public	Informational
PFC_FIXES	PFC revision level	Integer	Public	Informational
PFC_MAJOR	PFC major revision level	Integer	Public	Informational
PFC_MINOR	PFC minor revision	Integer	Public	Informational
PFC_NAME	PFC title	String	Public	Informational
PFC_BUILD_TIME	Build time	Time	Public	Informational
ids_debuglog	DataStore containing the debug log	n_ds	Public	Used by w_debuglog to display debugging messages
inv_sqlspy	Reference to SQL Spy debugging object	n_cst_sqlspy	Public	Internal

Events

N_cst_debug includes precoded events:

Constructor
Destructor

Constructor

Description: Creates an instance of ids_debuglog.

Usage: This event enables the debug log.

Destructor

Description: Performs clean up.

Usage: This event destroys all objects used by the debugging service.

Functions

N_cst_debug includes precoded functions:

of_ClearLog	of_PrintLog
of_GetAlwaysOnTop	of_SetAlwaysOnTop
of_IsLogOpen	of_SetDWProperty
of_Message	of_SetSQLSpy
of_OpenLog	

of_ClearLog

Description: Clears all data from the ids_debuglog DataStore.

Access: Public

Syntax: *instancename*.**of_ClearLog** ( )

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The w_debuglog window calls this function to clear the debugging log.

Examples

This example is from the w_debuglog cb_clear Clicked event:

```
gnv_app.inv_debug.of_ClearLog()
```

of_GetAlwaysOnTop

Description

Reports whether the w_debuglog window always displays on top of other windows.

Access

Public

Syntax

instancename.**of_GetAlwaysOnTop ( )**

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)

Return value

Boolean. Returns TRUE if the w_debuglog window is always on top and FALSE if it is not.

Examples

This example calls the of_GetAlwaysOnTop function:

```
Boolean   ib_ontop

ib_ontop = gnv_app.inv_debug.of_GetAlwaysOnTop()
...
```

of_IsLogOpen

Description

Reports whether the w_debuglog window is currently open.

Access

Public

Syntax

instancename.**of_IsLogOpen ( )**

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)

Return value

Boolean. Returns TRUE if the w_debuglog window is open and FALSE if it is not.

Examples

This example calls the of_IsLogOpen function:

```
Boolean   ib_open

ib_open = gnv_app.inv_debug.of_IsLogOpen()
...
```

of_Message

Description

Adds an entry to the debug log.

Access

Public

Syntax

instancename.**of_Message** (*message*)

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
message	String containing the text to be added to the debug log

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Message function:

```
Integer   li_return

li_return = gnv_app.inv_debug.of_Message &
    ("Login Error for: " + gnv_app.of_GetUserID())
```

of_OpenLog

Description

Opens or closes the w_debuglog window.

Access

Public

Syntax

instancename.**of_OpenLog** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)

Argument	Description
boolean	Boolean specifying whether to open (TRUE) or close (FALSE) the w_debuglog window

Return value

Integer. Returns the value of the PowerScript Open or Close function if the function succeeds or -1 if any argument's value is NULL.

Examples

This example calls the of_OpenLog function:

```
gnv_app.inv_debug.of_OpenLog(TRUE)
```

of_PrintLog

Description

Prints the debug log.

Access

Public

Syntax

instancename.**of_PrintLog ( )**

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_PrintLog function:

```
Integer   li_return

li_return = gnv_app.inv_debug.of_PrintLog()

IF li_return = -1 THEN
   MessageBox("Debug Log", "Error printing log")
ELSE
   gnv_app.of_GetFrame().SetMicroHelp &
      ("Log printed successfully")
END IF
```

of_SetAlwaysOnTop

Description

Controls whether the w_debuglog window always displays on top of other windows.

Access

Public

Syntax

instancename.**of_SetAlwaysOnTop** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
boolean	Boolean indicating whether to always display w_debuglog on top of other windows (TRUE) or to let other windows display on top of w_debuglog (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetAlwaysOnTop function:

```
...
gnv_app.inv_debug.of_SetAlwaysOnTop(FALSE)
...
```

of_SetDWProperty

Description

Enables or disables display of the DataWindow Properties window.

Access

Public

Syntax

instancename.**of_SetDWProperty** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
boolean	Boolean indicating whether to enable (TRUE) or disable (FALSE) display of the DataWindow Properties window

Return value

Integer. Returns 1 if the function succeeds, 0 if no action is taken, and -1 if an error occurs.

Usage

While running an application, you display the DataWindow Properties window by right-clicking over a DataWindow and selecting DataWindow Properties from the popup menu.

Examples

This example calls the of_SetDWProperty function:

```
gnv_app.of_SetDebug(TRUE)
gnv_app.inv_debug.of_SetDWProperty(TRUE)
```

of_SetSQLSpy

Description

Starts or stops the SQL Spy service.

Access

Public

Syntax

instancename.**of_SetSQLSpy** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
boolean	Boolean indicating whether to start (TRUE) or stop (FALSE) the SQL Spy service

Return value

Integer. Returns 1 if the function succeeds, 0 if no action is taken, and -1 if an error occurs.

Examples

This example calls the of_SetSQLSpy function:

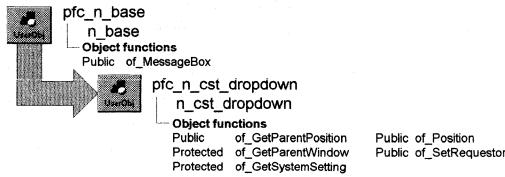
```
gnv_app.of_SetDebug(TRUE)
gnv_app.inv_debug.of_SetSQLSpy(TRUE)
```

n_cst_dropdown

Description

Dropdown service. The u_calculator and u_calendar objects use this object to control dropdown behavior when used with DataWindow columns and EditMasks.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

PFC objects use this object to control dropdown behavior. For usage information, see u_calculator and u_calendar.

See also

u_calculator
u_calendar

Instance variables

N_cst_dropdown includes instance variables:

Instance variable	Description	Data type	Access	Usage
iu_requestor	Requester object	UserObject	Protected	Control associated with this instance of the dropdown service
DWSTYLE_BOX	Constant set to 1	Integer	Protected	Internal
DWSTYLE_SHADOWBOX	Constant set to 2	Integer	Protected	Internal
DWSTYLE_LOWERED	Constant set to 3	Integer	Protected	Internal
DWSTYLE_RAISED	Constant set to 4	Integer	Protected	Internal
STYLE_BOX	Constant set to 5	Integer	Protected	Internal
STYLE_SHADOWBOX	Constant set to 6	Integer	Protected	Internal
STYLE_LOWERED	Constant set to 7	Integer	Protected	Internal

Instance variable	Description	Data type	Access	Usage
STYLE_RAISED	Constant set to 8	Integer	Protected	Internal
DW_HSPLITBAR_WIDTH	Constant set to 9	Integer	Protected	Internal
TAB_BORDER	Constant set to 10	Integer	Protected	Internal
MISC_XPOSITION	Constant set to 11	Integer	Protected	Internal
MISC_YPOSITION	Constant set to 12	Integer	Protected	Internal
DWMISC_XPOSITION	Constant set to 13	Integer	Protected	Internal
DWMISC_YPOSITION	Constant set to 14	Integer	Protected	Internal
DWDETAIL_HEIGHT	Constant set to 15	Integer	Protected	Internal
BORDER_CHECK	Constant set to 16	Integer	Protected	Internal

Functions

N_cst_dropdown contains precoded object functions:

of_GetParentPosition	of_Position
of_GetParentWindow	of_SetRequestor
of_GetSystemSetting	

of_GetParentPosition

Description

Retrieves the x and y coordinates for the parent object in relationship to the window on which it is placed.

Access

Public

Syntax

instancename.**of_GetParentPosition** (*control*, *x*, *y*)

Argument	Description
instancename	Instance name of n_cst_dropdown
control	DragObject whose parent position is returned
x	Integer into which the function places the parent's x coordinate (passed by reference)
y	Integer into which the function places the parent's y coordinate (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The of_Position function calls this function.

Examples

This example is from the of_Position function:

```
...
of_GetParentWindow(adrg_object, lw_parent)
IF IsNull(lw_parent) OR NOT IsValid(lw_parent) THEN
Return -1
END IF
of_GetParentPosition &
(adrg_object, li_parentx, li_parenty)
...
```

of_GetParentWindow

Description

Retrieves the window containing the current object.

Access

Protected

Syntax

instancename.**of_GetParentWindow** (*control*, *window*)

Argument	Description
instancename	Instance name of n_cst_dropdown
control	DragObject whose window is returned
window	Window variable into which the function places a reference to the window containing *control*. If there is no parent window, this argument returns NULL (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Position function:

```
...
of_GetParentWindow (adrg_object, lw_parent)
IF IsNull(lw_parent) OR NOT IsValid(lw_parent) THEN
Return -1
END IF
of_GetParentPosition &
(adrg_object, li_parentx, li_parenty)
...
```

of_GetSystemSetting

Description

Retrieves the platform-specific setting.

Access

Protected

Syntax

instancename.**of_GetSystemSetting** (*settingtype*)

Argument	Description
instancename	Instance name of n_cst_dropdown
settingtype	Integer or n_cst_dropdown constant specifying the setting to be returned: ♦ **1 or DWSTYLE_BOX** ♦ **2 or DWSTYLE_SHADOWBOX** ♦ **3 or DWSTYLE_LOWERED** ♦ **4 or DWSTYLE_RAISED** ♦ **5 or STYLE_BOX** ♦ **6 or STYLE_SHADOWBOX** ♦ **7 or STYLE_LOWERED** ♦ **8 or STYLE_RAISED** ♦ **9 or DW_HSPLITBAR_WIDTH** ♦ **10 or TAB_BORDER** ♦ **11 or MISC_XPOSITION** ♦ **12 or MISC_YPOSITION** ♦ **13 or DWMISC_XPOSITION** ♦ **14 or DWMISC_YPOSITION** ♦ **15 or DWDETAIL_HEIGHT** ♦ **16 or BORDER_CHECK**

Return value

Integer. Returns the platform-specific setting.

Usage

Internal.

Examples

This example is from the of_GetParentPosition function:

```
...
CHOOSE CASE luo_parent.BorderStyle
CASE StyleBox!
    li_border = of_GetSystemSetting(STYLE_BOX)
```

```
CASE StyleShadowBox!
   li_border = &
      of_GetSystemSetting(STYLE_SHADOWBOX)
CASE StyleLowered!
   li_border = &
      of_GetSystemSetting(STYLE_LOWERED)
CASE StyleRaised!
   li_border = &
      of_GetSystemSetting(STYLE_RAISED)
END CHOOSE
...
```

of_Position

Description

Updates the position of the requestor object based on the current position of the passed control.

Access

Public

Syntax

instancename.**of_Position** (*control*, *makevisible*)

Argument	Description
instancename	Instance name of n_cst_dropdown
control	DragObject containing the control used to calculate the new position
makevisible	Boolean indicating whether the dropdown object should be made visible after calculating a new position

Return value

Integer. Returns values as follows:

- 1 Success
- -1 An error occurred
- -2 The requested functionality is not supported
- -3 Unable to calculate a valid position

Examples

This example from the u_calendar of_DropDown function calls the of_Position function:

```
...
li_rc = inv_dropdown. of_Position &
(idrg_requestor, FALSE)
IF li_rc < 0 THEN Return -1
```

...

of_SetRequestor

Description

Associates this instance of n_cst_dropdown with a dropdown object (such as u_calendar).

Access

Public

Syntax

instancename.**of_SetRequestor** (*requestor*)

Argument	Description
instancename	Instance name of n_cst_dropdown
requestor	UserObject variable containing a reference to the dropdown object to associate with this instance of n_cst_dropdown

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetRequestor function:

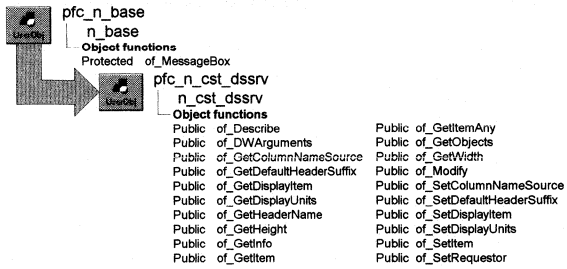
```
Integer li_rc

IF IsNull (ab_switch) THEN Return -1
  IF ab_switch THEN
    IF IsNull(inv_dropdown) OR NOT &
      IsValid(inv_dropdown) THEN
        inv_dropdown = CREATE n_cst_dropdown
        inv_dropdown.of_SetRequestor(this)
        li_rc = 1
    END IF
  ELSE
    IF IsValid (inv_dropdown) THEN
      DESTROY inv_dropdown
      li_rc = 1
    END IF
  END IF
Return li_rc
```

n_cst_dssrv

Description

Base DataStore service object.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Usage

Call the functions in this object to obtain basic information about a DataStore and its contents.

N_cst_dssrv is identical to n_cst_dwsrv, the base DataWindow service object.

FOR INFO For complete information on the instance variables, events, and functions in n_cst_dssrv, see n_cst_dwsrv on page 695.

Descendants

n_cst_dssrv_multitable
n_cst_dssrv_printpreview
n_cst_dssrv_report

See also

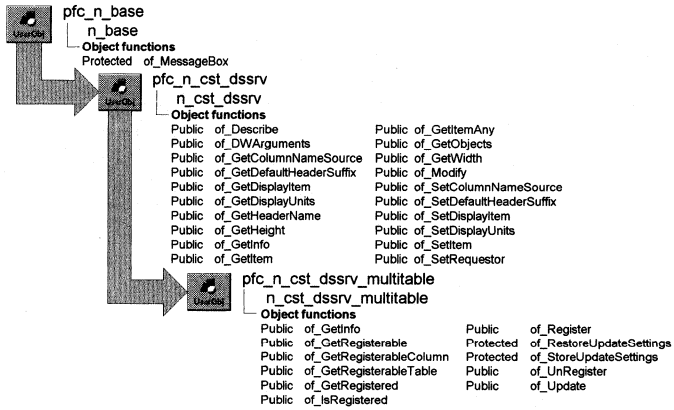
n_cst_dwsrv
n_ds

n_cst_dssrv_multitable

Description

A collection of functions that facilitate the updating of DataStores that contain rows from more than one table.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Usage

Use this object to handle update of multiple tables within a DataStore.

N_cst_dssrv_multitable is identical to n_cst_dwsrv_multitable, the DataWindow multitable update service object.

FOR INFO For complete information on the instance variables, events, and functions in n_cst_dssrv_multitable, see n_cst_dwsrv_multitable on page 832.

See also

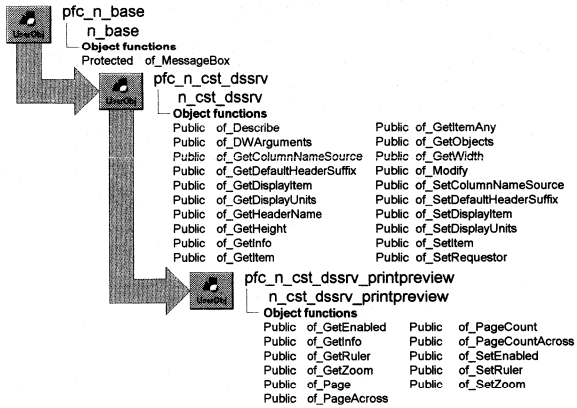
n_cst_dssrv
n_cst_dssrv_printpreview
n_cst_dssrv_report
n_cst_dwsrv_multitable
n_ds

n_cst_dssrv_printpreview

Description

Contains functions that allow you to control print preview.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Usage

Use this object to provide print preview functionality for DataStores.

N_cst_dssrv_printpreview is identical to n_cst_dwsrv_printpreview, the DataWindow print preview service object.

FOR INFO For complete information on the instance variables, events, and functions in n_cst_dssrv_printpreview, see n_cst_dwsrv_printpreview on page 844.

See also

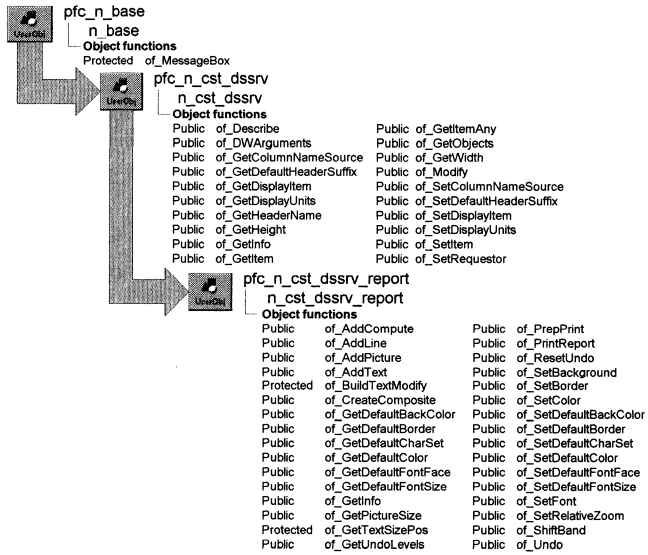
n_cst_dssrv
n_cst_dssrv_multitable
n_cst_dssrv_report
n_cst_dwsrv_printpreview
n_ds

n_cst_dssrv_report

Description

Contains functions that help you handle report DataStores and their printing.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Usage

Use this service to provide enhanced viewing and printing capabilities to DataStores that contain reports.

N_cst_dssrv_report is identical to n_cst_dwsrv_report, the DataWindow report service object.

FOR INFO For complete information on the instance variables, events, and functions in n_cst_dssrv_report, see n_cst_dwsrv_report on page 870.

See also

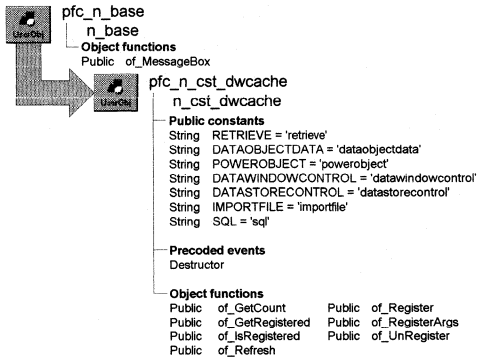
n_cst_dssrv
n_cst_dssrv_multitable
n_cst_dssrv_printpreview
n_cst_dwsrv_report
n_ds

n_cst_dwcache

Description

Caching service. This object is designed to contain data used in multiple places within an application. It provides a single data source, eliminating redundant database retrievals.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_ds
n_tr

Usage

To use caching:

1 Enable the caching service with the n_cst_appmanager of_SetDwCache function:

```
gnv_app.of_SetDwCache(TRUE)
```

2 Cache data by calling the of_Register function, passing different arguments depending on the data to be cached:

- To cache rows retrieved from the database via a DataWindow object, pass an identifier, a transaction object, the DataWindow object name, and arguments if any
- To cache rows retrieved from the database via a SQL statement, pass an identifier, a transaction object, and the SQL statement
- To cache rows in an array, pass an identifier, the DataWindow object name, and the data
- To cache rows from a DataWindow control, pass an identifier and the DataWindow control

- To cache rows from a DataStore, pass an identifier and the DataStore instance
- To cache rows from a file, pass an identifier and the filename

3 Call the of_GetRegistered function to access cached data. This example assumes an ids_datastore instance variable:

```
gnv_app.inv_dwcache.of_GetRegistered &
   ("d_emplist", ids_datastore)
ids_datastore.ShareData(dw_emplist)
```

4 Refresh the cache object as needed, by calling of_Refresh.

Structures

Structure	Field	Data type
os_cachelist	a_args[ ]	Any
	ds_obj	n_ds
	s_id	String
	s_method	String
	s_filename	String
	dw_originalcontrol	DataWindow
	ds_originalcontrol	DataStore
	po_originaldata[ ]	PowerObject
	s_originaldataobject	String
	tr_obj	n_tr

See also

n_cst_appmanager

Instance variables

N_cst_dwcache includes instance variables:

Instance variable	Description	Data type	Access	Usage
DATAOBJECTDATA	Constant that indicates original data location and refresh functionality	String	Public	Internal

Instance variable	Description	Data type	Access	Usage
POWEROBJECT	Constant that indicates original data location and refresh functionality	String	Public	Internal
RETRIEVE	Constant that indicates original data location and refresh functionality	String	Public	Internal
istr_cachelist	Os_cachelist array containing cached data	Os_cachelist	Protected	Internal
DATASTORECONTROL	Constant that indicates original data location and refresh functionality a	String	Public	Internal
DATAWINDOWCONTROL	Constant that indicates original data location and refresh functionality	String	Public	Internal
IMPORTFILE	Constant that indicates original data location and refresh functionality	String	Public	Internal
SQL	Constant that indicates original data location and refresh functionality	String	Public	Internal

Events

N_cst_dwcache includes one precoded event:

Destructor

Destructor

Description Destroys all cache objects.

Usage This event destroys all DataStores used by the caching service.

Functions

N_cst_dwcache includes precoded object functions:

of_GetCount	of_Register
of_GetRegistered	of_RegisterArgs
of_IsRegistered	of_UnRegister
of_Refresh	

of_GetCount

Description

Retrieves the number of objects currently registered.

Access

Public

Syntax

application.instancename.**of_GetCount** ( )

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)

Return value

Integer. Returns the number of objects currently registered.

Examples

This example calls the of_GetCount function:

```
Integer   li_registered

li_registered = gnv_app.inv_dwcache.of_GetCount()
...
```

of_GetRegistered

Accesses cached data. There are three syntaxes:

To access	Use
A specified DataStore	Syntax 1
The IDs for all DataStores	Syntax 2
All DataStores	Syntax 3

Syntax 1

To access a specified DataStore

Description

Returns a reference to the DataStore matching the passed ID.

Access

Public

Syntax

application.instancename.**of_GetRegistered** (*id, cache*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String specifying the cached DataWindow objects ID, as specified in the of_Register function
cache	N_ds variable into which the function places a reference to the cached data's DataStore (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when you know the ID and need to access a single item.

To access all IDs, use Syntax 2; to access all DataStores, use Syntax 3.

Examples

This example calls the of_GetRegistered function. This example assumes an ids_datastore instance variable:

```
gnv_app.inv_dwcache.of_GetRegistered &
    ("d_emplist", ids_datastore)
ids_datastore.ShareData(dw_emplist)
```

Syntax 2

To access all IDs

Description

Returns a reference to all IDs in the istr_cachelist array.

Access

Public

Syntax

application.instancename.**of_GetRegistered** (*idlist*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
idlist	String array into which the function places the cached IDs, as specified in the of_Register function (passed by reference)

Return value

Integer. Returns the number of elements in the *idlist* array if the function succeeds and -1 if an error occurs.

Usage

Call this function to access all IDs.

To access a single item, use Syntax 1; to access all DataStores, use Syntax 3.

Examples

This example calls the of_GetRegistered function:

```
String    idlist[ ]

gnv_app.inv_dwcache.of_GetRegistered &
    (idlist)
...
```

Syntax 3

To access all DataStores

Description

Returns a reference to all DataStores in the istr_cachelist array.

Access

Public

Syntax

application.instancename.**of_GetRegistered** (*dslist*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
dslist	N_ds array into which the function places the cached DataStores (passed by reference)

Return value

Integer. Returns the number of elements in the *dslist* array if the function succeeds and -1 if an error occurs.

Usage

Call this function to access all DataStores.

To access a single item, use Syntax 1; to access all IDs, use Syntax 2.

Examples

This example calls the of_GetRegistered function:

```
n_ds    dslist[ ]

gnv_app.inv_dwcache.of_GetRegistered &
    (dslist)
...
```

of_IsRegistered

Description

Reports whether a specified ID is registered with the DataWindow caching service.

Access

Public

Syntax

application.instancename.**of_IsRegistered** (*id*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String specifying the ID specified in the of_Register function

Return value

Integer. Returns 1 if *id* is registered, 0 if it is not, and -1 if an error occurs.

Examples

This example calls the of_IsRegistered function:

```
Integer   li_return

li_return = gnv_app.inv_dwcache.of_IsRegistered &
   ("d_emplist")
CHOOSE CASE li_return
CASE 1
   gnv_app.inv_dwcache.of_GetRegistered &
      ("d_emplist", ids_datastore)
   ids_datastore.ShareData(dw_emplist)
CASE 0
   gnv_app.inv_dwcache.of_Register &
      ("d_emplist", "d_emplist", SQLCA)
   gnv_app.inv_dwcache.of_GetRegistered &
      ("d_emplist", ids_datastore)
   ids_datastore.ShareData(dw_emplist)
CASE -1
   MessageBox("DW Caching", "Error")
END CHOOSE
```

of_Refresh

Description

Refreshes one or all DataStores registered with the caching service.

Access

Public

Syntax

application.instancename.**of_Refresh** ({ *id* })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id (optional)	Specifies the ID of the cached object to be refreshed

Return value

Long. Returns values as follows:

- **If you specified an ID** Returns the number of rows retrieved
- **If you did not specify an ID** Returns the number of DataStores that were refreshed

Usage

Call this function to reretrieve rows for one or all cached objects.

Must be able to refresh
Not all cached objects are refreshable. For example, the specified Transaction object may no longer be connected.

Examples

This example calls the of_Refresh function:

```
gnv_app.inv_dwcache.of_Refresh("d_emplist")
```

of_Register

Enables caching for a specified object. There are seven syntaxes:

To	Use
Cache data retrieved via a DataWindow object, optionally using retrieval arguments	Syntax 1
Cache data retrieved via a DataWindow object	Syntax 2
Cache data retrieved via a SQL statement	Syntax 3
Cache data from a DataWindow control	Syntax 4
Cache data from a DataStore	Syntax 5
Cache passed data	Syntax 6
Cache data from a file	Syntax 7

Protected versions of of_Register

There are also two protected versions of of_Register. These are for internal use.

Syntax 1

To cache data retrieved via a DataWindow object, optionally using retrieval arguments

Description

Retrieves rows from the database (using the specified DataWindow object) and caches them.

Access

Public

Syntax

application.instancename.**of_Register** (*id*, *dwobject* {, *trans* {, *arguments* {, *initialload* } } })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String identifying the cached DataWindow object
dwobject	String specifying the DataWindow object to be cached
trans (optional)	N_tr transaction object used to retrieve rows for *dwobject* If you do not specify this argument, *dwobject* must have been saved with the data
arguments (optional)	Twenty-element array of the Any data type specifying arguments for *dwobject*
initialload (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request

Return value

Integer. Returns 1 if the function succeeds, 0 if *dwobject* is already registered, -1 if an error occurs, -2 if there is an error with *trans*, and -3 if there is a retrieve error.

Usage

Call this function to cache data for a DataWindow object that uses retrieval arguments.

Examples

This example calls the of_Register function:

```
gnv_app.inv_dwcache.of_Register &
    ("d_emplist", "d_emplist", SQLCA)
...
```

Syntax 2 **To cache data retrieved via a DataWindow object**

Description

Retrieves rows from the database (using the specified DataWindow object) and caches them.

Access

Public

Syntax

application.instancename.**of_Register** (*id, dwobject* {, *trans* {, *initialload* } })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String identifying the cached DataWindow object
dwobject	String specifying the DataWindow object to be cached
trans (optional)	N_tr transaction object used to retrieve rows for *dwobject* If you do not specify this argument, *dwobject* must have been saved with the data
initialload (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request

Return value

Integer. Returns 1 if the function succeeds, 0 if *dwobject* is already registered, -1 if an error occurs, -2 if there is an error with *trans*, and -3 if there is a retrieve error.

Usage

Call this function to cache data for a DataWindow object that has no retrieval arguments.

Examples

This example calls the of_Register function:

```
gnv_app.inv_dwcache.of_Register &
    ("d_emplist", "d_emplist", SQLCA, TRUE)
...
```

Syntax 3 **To cache data retrieved via a SQL statement**

Description

Retrieves rows (using a specified SQL statement) and caches them.

Access

Public

Syntax

application.instancename.**of_Register** (*id, trans, sql* {, *initialload* })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String identifying the cached DataWindow object
trans	N_tr transaction object used to retrieve rows for *dwobject*
sql	String specifying the SQL SELECT statement used to create the cache
initialload (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request

Return value

Integer. Returns 1 if the function succeeds, 0 if *id* is already registered, and -1 if an error occurs.

Usage

Call this function to cache data for a DataWindow object, using SQL to retrieve rows from the database.

Examples

This example calls the of_Register function:

```
gnv_app.inv_dwcache.of_Register &
    ("sql_empall", SQLCA, &
      "SELECT * FROM employee")
...
```

Syntax 4

To cache data from a DataWindow control

Description

Caches rows from a passed DataWindow control.

Access

Public

Syntax

application.instancename.**of_Register** (*id, dwcontrol* {, *initialload* })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache The n_cst_appmanager default for this is inv_dwcache
id	String identifying the cached DataWindow object
dwcontrol	DataWindow control instance whose DataWindow object contains the data to be cached

Argument	Description
initialload (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request

Return value

Integer. Returns 1 if the function succeeds, 0 if *dwobject* is already registered, and -1 if an error occurs.

Usage

Call this function to cache data from a DataWindow control.

Examples

This example calls the of_Register function:

```
gnv_app.inv_dwcache.of_Register &
    ("d_emplist", dw_emplist, TRUE)
```

Syntax 5

To cache data from a DataStore

Description

Caches rows from a passed DataStore.

Access

Public

Syntax

application.instancename.**of_Register** (*id, datastore* {, *initialload* })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String identifying the cached DataWindow object
datastore	DataStore instance whose DataWindow object contains the data to be cached
initialload (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request

Return value

Integer. Returns 1 if the function succeeds, 0 if *id* is already registered, and -1 if an error occurs.

Usage

Call this function to cache data from a DataStore.

Examples

This example calls the of_Register function:

```
gnv_app.inv_dwcache.of_Register &
    ("d_emplist", ids_emplist, TRUE)
```

Syntax 6 **To cache passed data**

Description Caches passed data.

Access Public

Syntax *application.instancename*.**of_Register** (*id, dwobject, data*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String identifying the cached object
dwobject	String specifying the DataWindow object to be cached
data	PowerObject array of structures whose elements match *dwobject*. This array contains the data to be cached

Return value Integer. Returns 1 if the function succeeds, 0 if *dwobject* is already registered, and -1 if an error occurs.

Usage Call this function to cache external data for a DataWindow object.

Elements must match
Application errors will occur if the elements of the *data* structure do not exactly match the columns of *dwobject*.

The of_Refresh function does not affect data cached with this syntax.

Examples This example calls the of_Register function:

```
s_newdata   str_newdata[ ]

... // Logic to populate str_newdata
gnv_app.inv_dwcache.of_Register &
   ("d_emplist", "d_emplist", str_newdata)
```

Syntax 7 **To cache data from a file**

Description Caches data from a file.

Access Public

Syntax *application.instancename*.**of_Register** (*id, dwobject, filename* {, *initialload* })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	String identifying the cached object
dwobject	String specifying the DataWindow object to be cached
filename	String specifying the name of a tab-delimited file whose elements match *dwobject*
initialload (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request

Return value

Integer. Returns 1 if the function succeeds, 0 if *dwobject* is already registered, and -1 if an error occurs.

Usage

Call this function to cache data from a file.

Elements must match
Application errors will occur if the elements of the lines in *filename* do not exactly match the columns of *dwobject*.

The of_Refresh function does not affect data cached with this syntax.

To cache data from a database, use Syntax 1; to cache data stored with the DataWindow object, use Syntax 3.

Examples

This example calls the of_Register function:

```
gnv_app.inv_dwcache.of_Register &
   ("d_emplist", "d_emplist", "c:\temp\emplist.txt")
```

of_RegisterArgs

Description

Modifies retrieval arguments for a registered object.

Access

Public

Syntax

application.instancename.**of_RegisterArgs** (*id, arguments*)

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id	ID of the cached object
arguments	Twenty-element array of the Any data type specifying new arguments for the DataWindow object associated with *id*

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to establish new retrieval arguments for a previously cached DataWindow.

The number of arguments specified in *arguments* must match the number of arguments specified when the cached object was registered.

Examples

This example calls the of_RegisterArgs function:

```
Any la_args[20]

// Code to populate la_args
...
gnv_app.inv_dwcache.of_RegisterArgs &
   ("d_emplist", la_args)
```

of_UnRegister

Description

Removes one or all objects from the caching service.

Access

Public

Syntax

application.instancename.**of_UnRegister** ({ *id* })

Argument	Description
application	Global variable of type n_cst_appmanager (must be gnv_app)
instancename	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
id (optional)	ID of the DataWindow object to be removed from the cache

Return value

Integer. Returns the number of objects that were unregistered if the function succeeds and -1 if an error occurs.

Usage

If you do not specify *id*, the function unregisters all objects.

Examples

This example calls the of_UnRegister function:

```
gnv_app.inv_dwcache.of_UnRegister &
    ("d_emplist")
```

n_cst_dwsrv

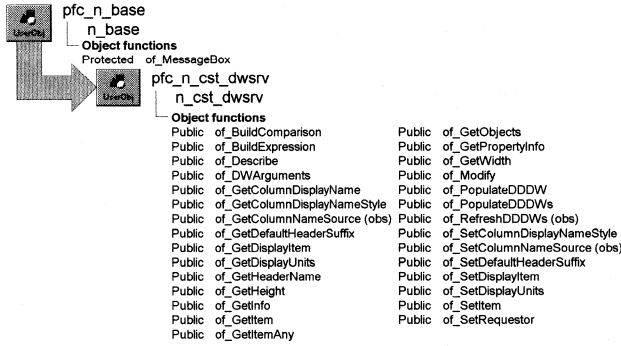
Description

Base DataWindow services object. N_cst_dwsrv serves as the ancestor object for all DataWindow services. In addition to being the ancestor for all other DataWindow services, u_dw instantiates this user object as inv_base.

N_cst_dssrv is the same as n_cst_dwsrv

N_cst_dssrv, the base DataStore service object is almost identical to n_cst_dwsrv except that it applies to DataStores instead of DataWindow controls. N_cst_dssrv has the same relationship to n_ds as n_cst_dwsrv has to u_dw. Except where noted, references to n_cst_dwsrv in the following discussions apply equally to n_cst_dssrv.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_string
n_cst_dwobjectattrib
n_cst_infoattrib
n_cst_propertyattrib
u_dw

Usage

Call the functions in this user object to obtain basic information about a DataWindow object and its contents.

To use this service:

1 Enable the service using the u_dw of_SetBase function:

```
dw_emp.of_SetBase(TRUE)
```

2 Call base DataWindow service functions as needed:

```
dw_emp.inv_base.of_SetDisplayUnits &
   ("selected employees")
dw_emp.inv_base.of_SetDisplayItem &
   ("the selected employee")
```

Descendants

n_cst_dwsrv_dropdownsearch
n_cst_dwsrv_filter
n_cst_dwsrv_find
n_cst_dwsrv_linkage
n_cst_dwsrv_multitable
n_cst_dwsrv_printpreview
n_cst_dwsrv_property
n_cst_dwsrv_querymode
n_cst_dwsrv_report
n_cst_dwsrv_reqcolumn
n_cst_dwsrv_resize
n_cst_dwsrv_rowmanager
n_cst_dwsrv_rowselection
n_cst_dwsrv_sort

See also

u_dw
n_cst_dssrv

Instance variables

N_cst_dwsrv includes instance variables:

Instance variable	Description	Data type	Access	Usage
DBNAME	Constant set to 1	Integer	Public	Use with of_SetColumnDisplayName Style
DEFAULT	Constant set to 0	Integer	Public	Use with of_SetColumnDisplayName Style
HEADER	Constant set to 2	Integer	Public	Use with of_SetColumnDisplayName Style
idw_requestor	The DataWindow control that owns the n_cst_dwsrv instance	u_dw	Protected	Used to access and control the associated DataWindow control

Instance variable	Description	Data type	Access	Usage
ii_source	Used by n_cst_dwsrv_querymode	Integer	Protected	Access
is_defaultheadersuffix	Default suffix for DataWindow column headers and labels (the default is _t)	String	Protected	Call of_SetDefaultHeaderSuffix and of_GetDefaultHeaderSuffix to access
is_displayitem	Controls message text displayed for a single row	String	Protected	Call of_SetDisplayItem and of_GetDisplayItem to access
is_displayunits	Controls message text displayed for multiple rows	String	Protected	Call of_SetDisplayUnits and of_GetDisplayUnits to access

Functions

N_cst_dwsrv includes precoded object functions:

of_BuildComparison	of_GetObjects
of_BuildExpression	of_GetPropertyInfo
of_Describe	of_GetWidth
of_DWArguments	of_Modify
of_GetColumnDisplayName	of_PopulateDDDW
of_GetColumnDisplayNameStyle	of_PopulateDDDWs
of_GetColumnNameSource	of_RefreshDDDWs
of_GetDefaultHeaderSuffix	of_SetColumnDisplayNameStyle
of_GetDisplayItem	of_SetColumnNameSource
of_GetDisplayUnits	of_SetDefaultHeaderSuffix
of_GetHeaderName	of_SetDisplayItem
of_GetHeight	of_SetDisplayUnits
of_GetInfo	of_SetItem
of_GetItem	of_SetRequestor
of_GetItemAny	

of_BuildComparison

Description

Returns a comparison string that can be used in a Find or Filter operation. This string does not contain the column specification, so it can be used with other columns.

Access

Public

Syntax

dwcontrol.instancename.**of_BuildComparison** (*row, column* {, *value* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
row	Long specifying the row number containing the column value
column	String specifying the name of the column containing the value
value (optional)	String specifying a value to be cast to the Any datatype. If you specify this argument, the function does not use *row* and *column*

Return value

Any. Returns the column's value cast to the Any data type if the function succeeds and ! if an error occurs.

Usage

Internal.

Examples

This example is from the n_cst_dwsrv_linkage pfc_PreDeleteRow event:

```
...
li_upperbound = UpperBound(is_keycols)
FOR li_i = 1 TO li_upperbound
   is_keycolsvalue[li_i] = of_BuildComparison &
     (al_row, is_keycols[li_i] )
NEXT
...
```

of_BuildExpression

Description

Creates a complete expression that can be used with a find or filter operation.

Access

Public

Syntax

dwcontrol.instancename.**of_BuildExpression** (*row, column* {, *operator* {, *value* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Argument	Description
row	Long specifying the row number containing the column value used in the comparison string
column	String specifying the column used in the comparison string
operator (optional)	String specifying the operator used in the comparison string (the default is =)
value (optional)	String specifying a value used in the comparison string. If you specify this argument, the function does not use *row* and *column* to determine the value

Return value

Any. Returns a comparison string if the function succeeds and ! if an error occurs.

Usage

Internal.

Examples

This example is from the n_cst_dwsrv_linkage of_ScrollDetails function:

```
...
ls_expression = of_BuildExpression  &
   (0, inv_linkargs.is_detailcolarg[li_i],  &
     '=', ls_value)
...
```

of_Describe

Describes DataWindow attributes. There are two syntaxes:

To	Use
Describe multiple columns or bands	Syntax 1
Describe a single column	Syntax 2

Syntax 1

To describe multiple columns or bands

Description

Describes the specified attribute for multiple columns or bands.

Access

Public

Syntax

dwcontrol.instancename.**of_Describe** (*returndescribe, attribute* {, *objecttype, band, visibleonly* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
returndescribe	N_cst_dwobjectattrib instance into which the function places the described attributes
attribute	String specifying the name of the attribute to describe
objecttype (optional)	String specifying the type of objects to describe. The default is *, which describes all types For more information on DataWindow types, see of_GetObjects on page 711
band (optional)	String specifying the band to describe (specify * to return all bands). If you specify *objecttype*, you must specify this argument. The default is *, which describes all bands. Valid bands are: ♦ Header ♦ Detail ♦ Footer ♦ Summary ♦ Header.*levelnumber* ♦ Trailer.*levelnumber*
visibleonly (optional)	Boolean indicating whether to describe visible columns only (TRUE) or all columns (FALSE). If you specify *objecttype*, you must specify this argument. The default is FALSE

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function as an alternative to multiple Describe functions.

Use Syntax 2 to describe a single column.

Examples

This example describes the border attribute for all visible columns in all bands:

```
n_cst_dwobjectattrib lnv_dwobjectattrib[ ]

dw_employee.inv_base.of_Describe &
   (lnv_dwobjectattrib, "border", "column", &
      "*", TRUE)
```

Syntax 2

To describe a single column

Description

Describes the specified attribute for a single column.

Access

Public

Syntax

dwcontrol.instancename.**of_Describe** (*returndescribe, attribute, columnname*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
returndescribe	N_cst_dwobjectattrib instance into which the function places the described attributes
attribute	String specifying the name of the attribute to describe
columnname	String specifying the column to describe

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function as an alternative to multiple Describe functions.

Use Syntax 1 to describe multiple columns.

Examples

This example describes the border attribute for the emp_id column:

```
n_cst_dwobjectattrib lnv_dwobjectattrib[ ]

dw_employee.inv_base.of_Describe &
    (lnv_dwobjectattrib, "border", "emp_id")
```

of_DWArguments

Determines DataWindow and DropDownDataWindow arguments. There are two syntaxes:

To	Use
Determine DropDownDataWindow arguments	Syntax 1
Determine DataWindow arguments	Syntax 2

Syntax 1

To access DropDownDataWindow arguments

Description

Determines if the passed dropdown DataWindow has retrieval arguments.

Access

Public

Syntax

dwcontrol.instancename.**of_DWArguments** (*dwchild, arguments, datatypes*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
dwchild	DataWindowChild variable specifying the DropDownDataWindow to be tested for retrieval arguments
arguments	String array into which the function places argument names (passed by reference)
datatypes	String array into which the function places argument data types (passed by reference)

Return value

Integer. Returns the number of elements in the *arguments* and *datatypes* arrays.

Usage

Internal.

Examples

This example calls the of_DWArguments function:

```
Long  ll_cnt, ll_max, ll_num_dddws
DataWindowChild  ldwc_dddws[ ]
String  ls_dddw_col, ls_dddw_name
String  ls_args[ ], ls_types[ ]
...
IF of_DWArguments(ldwc_dddws[ll_num_dddws], &
   ls_args, ls_types ) > 0 THEN
      idw_Requestor.Event pfc_RetrieveDDDW &
         (ls_dddw_col)
ELSE
   ldwc_dddws [ll_num_dddws].SetTransObject &
      (idw_Requestor.itr_object)
   ldwc_dddws [ll_num_dddws].Retrieve( )
END IF
```

Syntax 2

To access DataWindow arguments

Description

Determines if the passed DataWindow has retrieval arguments.

Access

Public

Syntax

dwcontrol.instancename.**of_DWArguments** (*arguments*, *datatypes*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
arguments	String array into which the function places argument names (passed by reference)
datatypes	String array into which the function places argument data types (passed by reference)

Return value

Integer. Returns the number of elements in the *arguments* and *datatypes* arrays.

Examples

This example calls the of_DWArguments function:

```
Integer   li_return, li_count
String    ls_args[ ], ls_types[ ]

li_return = dw_employee.inv_base.of_DWArguments &
   (ls_args, ls_types)
IF li_return > 0 THEN
   FOR li_count = 1 to li_return
      // processing logic
   NEXT
ELSE
   MessageBox("Error", "No DW arguments")
END IF
```

of_GetColumnDisplayName

Description

Retrieves the display name for a specified column.

Access

Public

Syntax

dwcontrol.instancename.**of_GetColumnDisplayName** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
column	String specifying the column name or integer specifying the column number

Return value

String. Returns the display name for the column.

Usage

PFC uses the database column name, the DataWindow column name, or the column header name for the display name. You set the display name with the of_SetColumnDisplayNameStyle function.

Examples

This example calls the of_GetColumnDisplayName function:

```
String  ls_name

ls_name =  &
  dw_1.inv_base.of_GetColumnDisplayName("emp_id")
MessageBox("Name", "Display name is " + ls_name)
```

of_GetColumnDisplayNameStyle

Description

Retrieves the DataWindow's column display name style.

Access

Public

Syntax

dwcontrol.instancename.**of_GetColumnDisplayNameStyle ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Integer. Returns values as follows:

- **0 or DEFAULT** DataWindow column name
- **1 or DBNAME** Database column name
- **2 or HEADER** Column header name

Examples

This example calls the of_GetColumnDisplayNameStyle function:

```
Integer  li_style

li_style =  &
   dw_1.inv_base.of_GetColumnDisplayNameStyle()
MessageBox("Style",  &
   "Display style is " + String(li_style))
```

of_GetColumnNameSource

Description

Obsolete. Call of_GetColumnDisplayNameStyle instead.

Access

Public

Syntax

dwcontrol.instancename.**of_GetColumnNameSource** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Integer. Returns the column name source:

- 0—DataWindow column names
- 1—Database column names
- 2—Column header names

of_GetDefaultHeaderSuffix

Description

Retrieves the default header suffix.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDefaultHeaderSuffix** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

String. Returns the default header suffix.

Usage

Call this function to access the default suffix used by DataWindow objects to designate headers.

Examples

This example calls the of_GetDefaultHeaderSuffix function:

```
String    ls_suffix

ls_suffix = inv_base.of_GetDefaultHeaderSuffix( )
MessageBox("DataWindow", "Header suffix is: " &
   + ls_suffix)
```

of_GetDisplayItem

Description

Retrieves the display name for single items in the DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDisplayItem** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

String. Returns the display name for single rows of the DataWindow.

Usage

Call this function to access the display name for individual rows of the DataWindow. PFC uses this name in error message text when confirming single-row deletion. For example, if this name is *this row*, the n_cst_dwsrv_rowmanager function displays Are you sure you want to delete this row?

Examples

This example calls the of_GetDisplayItem function:

```
String    ls_item

ls_item = inv_base.of_GetDisplayItem( )
MessageBox("DataWindow", "Item label is: " &
   + ls_item)
```

of_GetDisplayUnits

Description

Retrieves the display name for multiple rows in the DataWindow. N_cst_dwsrv maintains this as the is_displayunits instance variable.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDisplayUnits** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

String. Returns the display name for multiple rows of the DataWindow.

Usage

Call this function to access the display name for the rows of the DataWindow. PFC uses this name in error message text when confirming multiple row deletion. For example, if this name is *rows*, the n_cst_dwsrv_rowmanager function displays Are you sure you want to delete these rows?

Examples

This example calls the of_GetDisplayUnits function:

```
String    ls_units

ls_units = inv_base.of_GetDisplayUnits( )
MessageBox("DataWindow", "Unit label is: " &
   + ls_units)
```

of_GetHeaderName

Description

Returns a header name for the passed DataWindow column name.

Access

Public

Syntax

dwcontrol.instancename.**of_GetHeaderName** (*columnname* {, *suffix* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
columnname	String specifying the DataWindow column name
suffix (optional)	String specifying the suffix the function uses to find the header name. The default is _t

Return value

String. Returns a formatted DataWindow column heading (with underscores, carriage returns, line feeds, and quotes removed).

Usage

The Sort and Filter dialog boxes used by the n_cst_dwsrv_querymode, n_cst_dwsrv_filter, and n_cst_dwsrv_sort services call this function to determine column header text.

Examples

This example calls the of_GetHeaderName function:

```
String    ls_colhead, ls_columns

ls_colhead = &
idw_dw.inv_sort.of_GetHeaderName &
   (ls_columns)
```

of_GetHeight

Description

Retrieves the height of the DataWindow object in the associated DataWindow control.

Access

Public

Syntax

dwcontrol.instancename.**of_GetHeight** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Long. Returns the height of the DataWindow object in the unit of measure specified for the DataWindow.

Examples

This example, from a window Open event, returns the height of the dw_detail DataWindow control:

```
Long   ll_height

ll_height = dw_detail.of_GetHeight( )
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
infoobject	n_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_base.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetItem

Description

Retrieves the formatted text of any DataWindow column, regardless of the column's data type. The returned text includes formats, editmasks, and display values, if any.

Access

Public

Syntax

dwcontrol.instancename.**of_GetItem** (*row*, *column* {, *buffer*, *originalvalue* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
row	Long specifying the row containing the column to be returned
column	String or integer identifying the column to be returned
buffer (optional)	DWBuffer enumerated data type optionally specifying the DataWindow buffer from which to return the value. The default is Primary!
originalvalue (optional)	Boolean specifying whether to return the original value (TRUE) or the current DataWindow value (FALSE). The default is FALSE. IF you specify *buffer*, you must also specify this argument

Return value

String. Returns the formatted text for the specifying row and column.

Examples

This example calls the of_GetItem function:

```
String    ls_string

IF al_row <= 0 THEN Return ""
/* Get the string value */
IF IsValid(idw_master) THEN
   IF IsValid(idw_master.inv_linkage) THEN
      ls_string = &
```

```
            idw_master.inv_linkage.of_GetItem &
               (al_row, as_column)
      ELSE
         Return ""
      END IF
   ELSE
      Return ""
   END IF
   ...
```

of_GetItemAny

Description

Returns a column's value, cast to the Any data type.

Access

Public

Syntax

dwcontrol.instancename.**of_GetItemAny** (*row*, *column* {, *buffer*, *originalvalue* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
row	Long specifying the row number of the value to be retrieved
column	String or integer specifying the name of the column for the value to be retrieved
buffer (optional)	String specifying the name of the DataWindow buffer from which to get the column's data value. The default is Primary!
originalvalue (optional)	Boolean specifying whether to return the originally retrieved database value (TRUE) or to return the current value (FALSE). The default is FALSE

Return value

Any. Returns the column value, cast to the Any data type.

Examples

This example is from the n_cst_dwsrv_linkage of_RetrieveDetails function:

```
Integer   li_num_links, li_i, li_rc
Any ls_args[20]
Any la_detail

li_num_links = &
   UpperBound(inv_linkargs.is_mastercolarg)
```

```
FOR li_i = 1 to li_num_links
   /* Get the column data value from the master */
   IF IsValid(idw_master) THEN
      IF IsValid(idw_master.inv_linkage) THEN
         ls_args[li_i] = &
            idw_master.inv_linkage.of_GetItemAny &
               (al_row, &
                  inv_linkargs.is_mastercolarg[li_i])
      ELSE
         Return -1
      END IF
   ELSE
      Return -1
   END IF
...
```

of_GetObjects

Description

Parses the list of objects contained in the associated DataWindow, placing their names into a string array passed by reference and returning the number of names in the array.

Access

Public

Syntax

dwcontrol.instancename.**of_GetObjects** (*objectlist* {, *objecttype*, *band*, *visibleonly* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
objectlist	String array into which the function places object names (passed by reference)

Argument	Description
objecttype (optional)	String specifying the type of objects to return (specify * to return all object types). The default is to return all object types. Valid types are: Bitmap Column Compute (for computed field) Graph Line Ellipse (for oval) Rectangle Report Roundrectangle Tableblob Text
band (optional)	String specifying the band from which to return objects (specify * to return all bands). The default is to return all bands. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
visibleonly (optional)	Boolean specifying whether to return visible objects only (TRUE) or to return all objects (FALSE). The default is FALSE

Return value

Integer. Returns the number of elements in the *objectlist* array.

Usage

Call this function to access information on the objects contained in the DataWindow object associated with the current DataWindow control.

You can control the objects returned by *type*, *band*, and *visibleonly*. Specifying an asterisk (*) for either *objecttype* or *band* means that you want all occurrences of objects across type or band. Specifying TRUE for *visibleonly* means that you only want visible objects; FALSE returns all objects.

Examples

This example calls the of_GetObjects function:

```
Integer   li_size
String    ls_objects[ ]

li_size = dw_employee.inv_base.of_GetObjects &
   (ls_objects[ ], "*", "detail", TRUE)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
propertyobject	n_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information:

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_base.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetWidth

Description

Retrieves the width (in the units specified for the DataWindow object) of the associated DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetWidth** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Long. Returns the DataWindow width in the units specified for the DataWindow object.

Examples

This example uses the of_GetWidth function to determine the width of the dw_employee DataWindow:

```
Long   ll_width

ll_width = dw_employee.inv_base.of_GetWidth( )
```

of_Modify

Modifies DataWindow attributes. There are two syntaxes:

To	Use
Modify multiple columns or bands	Syntax 1
Modify a single column	Syntax 2

Syntax 1

To modify multiple columns or bands

Description

Modifies the specified attribute for multiple columns or bands.

Access

Public

Syntax

dwcontrol.instancename.**of_Modify** (*attribute, value* {, *objecttype, band, visibleonly* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
attribute	String specifying the name of the attribute to be modified
value	String specifying the new value for the attribute
objecttype (optional)	String specifying the type of objects to modify. The default is *, which modifies all types FOR INFO For more information on DataWindow types, see of_GetObjects on page 711

Argument	Description
band (optional)	String specifying the band to modify (specify * to return all bands). The default is *, which modifies all bands. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
visibleonly (optional)	Boolean indicating whether to modify visible columns only (TRUE) or all columns (FALSE). The default is FALSE

Return value

String. Returns the string from the Modify function. An empty string indicates success.

Usage

Call this function as an alternative to multiple Modify functions.

Use Syntax 2 to modify a single column.

Examples

This example changes the border for all columns in all bands to 3D lowered (5):

```
String    ls_return

ls_return = dw_employee.inv_base.of_Modify &
    ("border", "5", "column", "*", TRUE)
```

Syntax 2

To modify a single column

Description

Modifies the specified attribute for a single column.

Access

Public

Syntax

dwcontrol.instancename.**of_Modify** (*attribute*, *value*, *columnname*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
attribute	String specifying the name of the attribute to be modified
value	String specifying the new value for the attribute
columnname	String specifying the column to modify

Return value

String. Returns the string from the Modify function. An empty string indicates success.

Usage

Call this function as an alternative to multiple Modify functions.

Use Syntax 1 to modify multiple columns.

Examples

This example changes the border for the emp_id column to 3D lowered (5):

```
String    ls_return

ls_return = dw_employee.inv_base.of_Modify &
   ("border", "5", "emp_id")
```

of_PopulateDDDW

Description

Populates the specified DropDownDataWindow column.

Access

Public

Syntax

dwcontrol.instancename.**of_PopulateDDDW** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
column	String specifying the column name or integer specifying the column number

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function retrieves rows for DropDownDataWindows without retrieval arguments. For a DropDownDataWindow that uses retrieval arguments, you must code a Retrieve function in the u_dw pfc_RetrieveDDDW event.

Not in n_cst_dssrv
This function is not implemented in n_cst_dssrv.

Examples

This example calls the of_PopulateDDDW function:

```
...
dw_1.inv_base.of_PopulateDDDW("dept_id")
...
```

of_PopulateDDDWs

Description

Populates all DropDownDataWindows in a DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_PopulateDDDWs** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function retrieves rows for DropDownDataWindows without retrieval arguments. For a DropDownDataWindow that uses retrieval arguments, you must code a Retrieve function in the u_dw pfc_RetrieveDDDW event.

Not in n_cst_dssrv
This function is not implemented in n_cst_dssrv.

Examples

This example calls the of_PopulateDDDWs function:

```
...
dw_1.inv_base.of_PopulateDDDWs()
...
```

of_RefreshDDDWs

Description

Obsolete. Call of_PopulateDDDWs instead.

Access

Public

Syntax

dwcontrol.instancename.**of_RefreshDDDWs** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Integer. Returns the number of DropDownDataWindows found and refreshed.

of_SetColumnDisplayNameStyle

Description

Specifies the DataWindow's column display name style.

Access

Public

Syntax

dwcontrol.instancename.**of_SetColumnDisplayNameStyle** (*style*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
style	Integer or n_cst_dwsrv constant specifying the display style: ♦ **0 or DEFAULT** DataWindow column name ♦ **1 or DBNAME** Database column name ♦ **2 or HEADER** Column header name

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to specify the column labels and headings used by the n_cst_dwsrv_querymode, n_cst_dwsrv_filter, and n_cst_dwsrv_sort services.

You typically call this function just after instantiating the service.

Examples

This example calls the of_SetColumnDisplayNameStyle function:

```
...
dw_1.inv_base.of_SetColumnDisplayNameStyle  &
 (dw_1.inv_base.HEADER)
```

of_SetColumnNameSource

Description

Obsolete. Call of_SetColumnDisplayNameStyle instead.

Access

Public

Syntax

dwcontrol.instancename.**of_SetColumnNameSource** (*source*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Argument	Description
source	Integer specifying the column name source: ♦ 0—DataWindow column names (default) ♦ 1—Database column names ♦ 2—DataWindow column header names

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_SetDefaultHeaderSuffix

Description

Specifies the default suffix that your application uses in DataWindow objects to designate headers.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultHeaderSuffix** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetDefaultHeaderSuffix function:

```
of_SetDefaultHeaderSuffix("_h")
```

of_SetDisplayItem

Description

Specifies the display name for single items in the DataWindow. N_cst_dwsrv maintains this as the is_displayitem instance variable.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDisplayItem** (*itemname*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
itemname	String specifying the display name for single rows of the DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to specify the display name for individual rows of the DataWindow. PFC uses this name in error message text when confirming single-row deletion. For example, if you specify *this row*, the n_cst_dwsrv_rowmanager function displays Are you sure you want to delete this row?

Examples

This example calls the of_SetDisplayItem function:

```
...
inv_base.of_SetDisplayItem("the selected row")
...
```

of_SetDisplayUnits

Description

Specifies the display name for multiple rows in the DataWindow. N_cst_dwsrv maintains this as the is_displayunits instance variable.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDisplayUnits** (*units*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
units	String specifying the display name for multiple rows of the DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to specify the display name for the rows of the DataWindow. PFC uses this name in error message text when confirming multiple row deletion. For example, if this name is *rows*, the n_cst_dwsrv_rowmanager function displays Are you sure you want to delete these rows?

Examples

This example calls the of_SetDisplayUnits function:

```
...
inv_base.of_SetDisplayUnits("selected rows")
...
```

of_SetItem

Description

Sets values in a specified row and column.

Access

Public

Syntax

dwcontrol.instancename.**of_SetItem** (*row*, *column*, *value*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
row	Long specifying the row containing the column to be updated
column	String or integer identifying the column to be updated. The function automatically converts this into the appropriate data type
value	String containing the new value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to update a DataWindow column when you aren't sure of the column's data type.

Examples

This example calls the of_SetItem function to blank out the emp_id column for the current row:

```
Integer   li_return
Long      ll_current

ll_current = dw_employee.GetRow()
li_return = inv_base.of_SetItem &
   (ll_current, "emp_id", "")
```

of_SetRequestor

Description

Associates a DataWindow service with a DataWindow control.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRequestor** (*requestor*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
requestor	U_dw-based DataWindow control to be associated with a service

Return value

None

Usage

The u_dw of_Set*servicename* functions call this function.

Examples

The u_dw of_SetReport function calls the of_SetRequestor function to associate the reporting service with u_dw-based DataWindow:

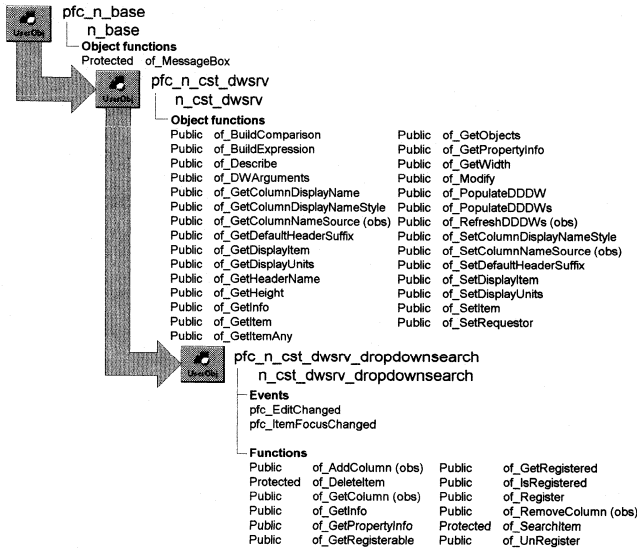
```
IF NOT IsValid (inv_Report) THEN
   inv_Report = CREATE n_cst_dwsrv_report
   inv_Report.of_SetRequestor(This)
END IF
```

n_cst_dwsrv_dropdownsearch

Description

Scrolls a DropDownDataWindow or DropDownListBox automatically, filling in the field as the user types.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

Use this object to add code to u_dw that provides automatic scrolling in DropDownDataWindows.

To use this service:

1 Enable the service by calling the u_dw of_SetDropDownSearch function.

```
dw_emp.of_SetDropDownSearch(TRUE)
```

2 Specify the DropDownDataWindow column for which the service is enabled:

```
dw_emp.inv_dropdownsearch.of_Register("dept_id")
```

3 In the DataWindow control's EditChanged event, add a call to the n_cst_dwsrv_dropdownsearch pfc_EditChanged event:

```
inv_dropdownsearch.Event pfc_EditChanged &
   (row, dwo, data)
```

4 In the DataWindow control's ItemFocusChanged event, add a call to the n_cst_dwsrv_dropdownsearch pfc_ItemFocusChanged event:

```
inv_dropdownsearch.Event pfc_ItemFocusChanged &
   (row, dwo)
```

Structures

Structure	Field	Data type
os_columns	dwc_object	DataWindowChild
	s_columnname	String
	s_editstyle	String

See also

n_cst_dwsrv
u_ddlb
u_ddplb
u_dw

Instance variables

N_cst_dwsrv_dropdownsearch includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_performsearch	Indicates whether the current column is a dropdown DataWindow	Boolean	Protected	Internal
ii_currentindex	Current index value	Integer	Protected	Internal
is_textprev	Previously typed text	String	Protected	Internal
istr_columns[ ]	Column and text information for dropdown DataWindows	os_columns	Protected	Internal

Events

N_cst_dwsrv_dropdownsearch includes precoded events:

pfc_EditChanged

pfc_ItemFocusChanged

pfc_EditChanged

Description

Scrolls the column to the first row that matches the typed characters.

Syntax

dwcontrol.instancename.EVENT **pfc_EditChanged** (*row*, *dwobject*, *data*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
row	Long containing the number of the row for the item whose value has been changed. From within the pfc_EditChanged event, access this value through the *al_row* argument (passed by reference)
dwobject	DWObject containing a reference to the column for the item whose value has been changed. *Dwobject* is a reference to the column object, not the name of the column. From within the pfc_EditChanged event, access this value through the *adwo_obj* argument (passed by reference)
data	String containing the new data the user has specified for the item. From within the pfc_EditChanged event, access this value through the *as_data* argument (passed by reference)

Usage

Add code to the DataWindow's EditChanged event to call this event.

Examples

This example calls the pfc_EditChanged event from the DataWindow's EditChanged event:

```
inv_dropdownsearch.Event pfc_EditChanged &
    (row, dwo, data)
```

pfc_ItemFocusChanged

Description

Initializes fields in the istr_columns structure array.

Syntax

dwcontrol.instancename.EVENT **pfc_ItemFocusChanged** (*row*, *dwobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
row	Long containing the number of the row. From within the pfc_ItemFocusChanged event, access this value through the *al_row* argument
dwobject	DWObject containing a reference to the column containing the item whose value has been changed. *Dwobject* is a reference to the column object, not the name of the column. From within the pfc_ItemFocusChanged event, access this value through the *adwo_object* argument (passed by reference)

Usage

This event executes when item focus changes. You must add code to the DataWindow's ItemFocusChanged event to call this event.

Examples

This example calls the pfc_ItemFocusChanged event from the DataWindow's ItemFocusChanged event:

```
inv_dropdownsearch.Event pfc_ItemFocusChanged &
    (row, dwo)
```

Functions

N_cst_dwsrv_dropdownseach contains precoded object functions:

of_AddColumn	of_GetRegistered
of_DeleteItem	of_IsRegistered
of_GetColumn	of_Register
of_GetInfo	of_RemoveColumn
of_GetPropertyInfo	of_SearchItem
of_GetRegisterable	of_UnRegister

of_AddColumn

Description

Obsolete. Call of_Register instead.

Access

Public

Syntax

dwcontrol.instancename.**of_AddColumn** ({ *column* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
column (optional)	String specifying the dropdown DataWindow column to add to the istr_columns array. This column should be a dropdown DataWindow and contain string data. If you do not specify this argument, the function adds all dropdown DataWindows to the istr_columns array

Return value

Integer. Returns the number of dropdown DataWindow columns added if the function succeeds, 0 if the column was not added, and -1 if an error occurs.

of_DeleteItem

Description

Removes an item from the istr_columns array and then compresses the array.

Access

Protected

Syntax

dwcontrol.instancename.**of_DeleteItem** (*index*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
index	Integer indicating which element of the istr_columns array to remove

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_DeleteItem function:

```
...
of_DeleteItem(li_index)
...
```

of_GetColumn

Description

Obsolete. Call of_GetRegistered instead.

Access

Public

Syntax

dwcontrol.instancename.**of_GetColumn** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
columns	String array into which the function places the dropdown DataWindow column names (passed by reference)

Return value

Integer. Returns the number of elements in the *columns* array.

of_GetInfo

Description

Retrieves dropdown search service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
infoobject	n_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_dropdownsearch.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the dropdown search service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
propertyobject	n_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information:

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_dropdownsearch.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegisterable

Description

Retrieves an array of registerable columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterable** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)

Argument	Description
columns	String array into which the function places a list of registerable columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Usage

A registerable column has the DropDownDataWindow or DropDownListBox edit style and has a display value of type Char.

Examples

This example calls the of_GetRegisterable function:

```
String   ls_columns[ ]
Integer  li_count, li_loop
li_loop =  &
   dw_1.inv_dropdownsearch.of_GetRegisterable  &
     (ls_columns)
lb_1.Reset()
IF li_loop <> 0 THEN
  FOR li_count = 1 to li_loop
    lb_1.AddItem(ls_columns[li_count])
  NEXT
ELSE
  lb_1.AddItem("No Registerable columns")
END IF
```

of_GetRegistered

Description

Retrieves an array of registered columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegistered** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
columns	String array into which the function places a list of registered columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Examples

This example calls the of_GetRegistered function:

```
String   ls_columns[ ]
Integer  li_count, li_loop

li_loop =  &
   dw_1.inv_dropdownsearch.of_GetRegistered  &
     (ls_columns)
lb_1.Reset()
IF li_loop <> 0 THEN
  FOR li_count = 1 to li_loop
    lb_1.AddItem(ls_columns[li_count])
  NEXT
ELSE
  lb_1.AddItem("No Registered columns")
END IF
```

of_IsRegistered

Description

Reports whether a column is registered.

Access

Public

Syntax

dwcontrol.instancename.**of_IsRegistered** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
column	String specifying the column to be checked

Return value

Boolean. Returns TRUE if *column* is registered and FALSE if it is not.

Examples

This example calls the of_IsRegistered function:

```
IF NOT dw_1.inv_dropdownsearch.of_IsRegistered  &
  ("dept_id") THEN
    dw_1.inv_dropdownsearch.of_Register("dept_id")
END IF
```

of_Register

Description

Registers one or all DropDownDataWindow and DropDownListBox columns with the dropdown search service.

Access

Public

Syntax

dwcontrol.instancename.**of_Register** ({ *column* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
column (optional)	String specifying the column to be registered. If you omit this argument, the function registers all eligible columns

Return value

Integer. Returns the number of columns registered if the function succeeds and -1 if an error occurs.

Usage

A registerable column has the DropDownDataWindow or DropDownListBox edit style and has a display value of type Char.

Examples

This example calls the of_Register function:

```
IF NOT dw_1.inv_dropdownsearch.of_IsRegistered  &
  ("dept_id") THEN
    dw_1.inv_dropdownsearch.of_Register("dept_id")
END IF
```

of_RemoveColumn

Description

Obsolete. Call of_UnRegister instead.

Access

Public

Syntax

dwcontrol.instancename.**of_RemoveColumn** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
column	String specifying the dropdown DataWindow column to remove from the istr_columns array

Return value

Integer. Returns the number of columns in the istr_columns array if the function succeeds, 0 if *column* is not found, and -1 if an error occurs.

of_SearchItem

Description

Searches the istr_columns array for the specified column name.

Access

Protected

Syntax

dwcontrol.instancename.**of_SearchItem** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
column	String specifying the column to be searched for

Return value

Integer. Returns the istr_columns array element containing *column*, 0 if *column* is not in istr_columns, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_ItemFocusChanged event:

```
...
li_index = of_SearchItem(ls_dwcolname)
...
```

of_UnRegister

Description

Removes one or all columns from the list of registered columns.

Access

Public

Syntax

dwcontrol.instancename.**of_UnRegister** ({ *column* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
column (optional)	String specifying the name of the column to unregister

Return value

Integer. Returns 1 if the function succeeds, 0 if *column* is not registered, and -1 if an error occurs.

Examples

This example calls the of_UnRegister function:

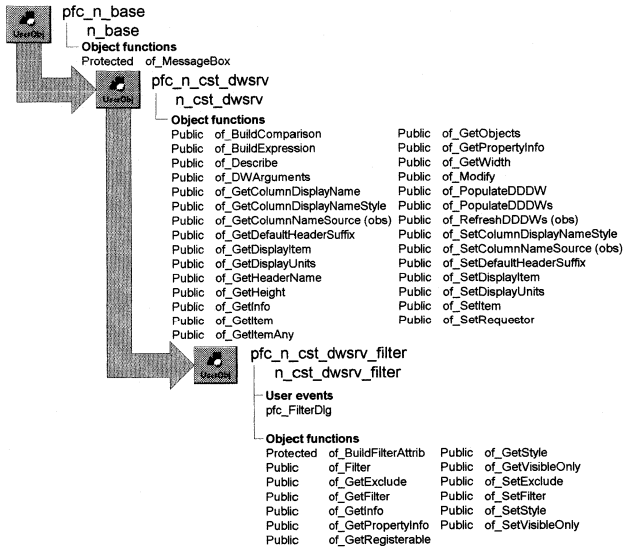
```
dw_1.inv_dropdownsearch.of_UnRegister("dept_id")
```

n_cst_dwsrv_filter

Description

Filter service. Provides functions that make it easy for you to add DataWindow filter functionality to an application. This service allows you to choose among three styles of filter dialog boxes.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_filterattrib
n_cst_infoattrib
n_cst_propertyattrib
n_cst_returnattrib
w_filterextended
w_filtersimple

Usage

Use this service to add filter capabilities to your DataWindow. For example, you might add a menu items that calls the u_dw pfc_FilterDlg event.

To use this service:

1 Enable the service using the u_dw of_SetFilter function:

```
dw_emp.of_SetFilter(TRUE)
```

2 (Optional) Specify whether filter dialog boxes should display database column names, DataWindow column names, or column header names, by calling the n_cst_dwsrv of_SetColumnDisplayNameStyle function:

```
dw_emp.inv_filter.of_SetColumnDisplayNameStyle(2)
```

3 (Optional) Specify the filter dialog box style by calling the of_SetStyle function:

```
dw_emp.inv_filter.of_SetStyle  &
   (dw_emp.inv_filter.EXTENDED)
```

4 Call the pfc_FilterDlg event:

```
dw_emp.inv_filter.Event pfc_FilterDlg()
```

This displays the filter dialog box.

See also

n_cst_dwsrv_sort
u_dw

Instance variables

N_cst_dwsrv_filter includes instance variables:

Instance variable	Description	Data type	Access	Usage
EXTENDED	Constant set to 1	Integer	Public	Use with of_SetStyle to specify filter dialog box style
ib_visibleonly	Controls whether Filter dialog boxes display nonvisible columns	Boolean	Protected	Internal
ii_style	Tracks filter style	Integer	Protected	Internal
is_excludecolumns[ ]	Tracks excluded columns	String	Protected	Internal
is_filterin	Filter the user specified	String	Protected	Internal
is_filterout	DataWindow filter	String	Protected	Internal
SIMPLE	Constant set to 2	Integer	Public	Use with of_SetStyle to specify filter dialog box style

Events

N_cst_dwsrv_filter contains a precoded user event:

pfc_FilterDlg

pfc_FilterDlg

Description: Displays the filter dialog box and filters the DataWindow.

Return value: Integer. Returns 1 if the function succeeds, 0 if the user canceled out of the dialog box, and -1 if an error occurs.

Usage: This event is called when the user selects View>Filter from the menu bar of a menu that descends from PFC's m_master menu.

Functions

N_cst_dwsrv_filter contains precoded object functions:

of_BuildFilterAttrib	of_GetStyle
of_Filter	of_GetVisibleOnly
of_GetExclude	of_SetExclude
of_GetFilter	of_SetFilter
of_GetInfo	of_SetStyle
of_GetPropertyInfo	of_SetVisibleOnly
of_GetRegisterable	

of_BuildFilterAttrib

Description: Prepares the filter information required by the filter dialog boxes.

Access: Protected

Syntax: *dwcontrol.instancename.***of_BuildFilterAttrib** (*filterattrib*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)

Argument	Description
filterattrib	N_cst_filterattrib instance into which the function places filter information

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_BuildFilterAttrib function:

```
...
IF of_BuildFilterAttrib(lnv_filterattrib) <> 1 THEN
   Return -1
END IF
...
```

of_Filter

Description

Filters a DataWindow and calls the u_dw pfc_RowChanged event to indicate that the current row may have changed.

Access

Public

Syntax

dwcontrol.instancename.**of_Filter** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function after calling the of_SetFilter function to filter a DataWindow.

Examples

This example calls the of_Filter function after calling the of_SetFilter function. Because of_SetFilter is passed a NULL argument, it displays a filter dialog box, as specified in the of_SetStyle function:

```
String    ls_null

SetNull(ls_null)
dw_emplist.inv_filter.of_SetStyle  &
   (dw_emplist.inv_filter.SIMPLE)
dw_emplist.inv_filter.of_SetFilter(ls_null)
```

```
dw_emplist.inv_filter.of_Filter( )
```

of_GetExclude

Description

Retrieves the list of excluded columns as set by the of_SetExclude function.

Access

Public

Syntax

dwcontrol.instancename.**of_GetExclude** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
columns	String array into which the function places the excluded columns (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetExclude function:

```
String    ls_exclude[ ]
String    ls_list
Integer   li_count

dw_emplist.inv_filter.of_GetExclude(ls_exclude)
For li_count = 1 to UpperBound(ls_exclude)
   ls_list = ls_list + ls_exclude[li_count] + ", "
NEXT
MessageBox("Filter Service", &
   "Excluded columns are: " + ls_list)
```

of_GetFilter

Description

Retrieves the current filter.

Access

Public

Syntax

dwcontrol.instancename.**of_GetFilter** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)

Return value

String. Returns the current filter.

Examples

This example calls the of_GetFilter function:

```
String    ls_filter

ls_filter = &
   dw_emplist.inv_filter.of_GetFilter()
MessageBox("Filter Service", "Filter is: " &
   + ls_filter)
```

of_GetInfo

Description

Retrieves filter service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_filter.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description: Retrieves information about the filter service's properties.

Access: Public

Syntax: *dwcontrol.instancename*.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: The DataWindow properties window calls this function to access service information.

Examples: This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_filter.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegisterable

Description: Retrieves an array of registerable columns.

Access: Public

Syntax: *dwcontrol.instancename*.**of_GetRegisterable** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)

Argument	Description
columns	String array into which the function places a list of registerable columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Usage

You can register all columns, including computed columns.

Examples

This example calls the of_GetRegisterable function:

```
String   ls_columns[ ]
Integer  li_count, li_loop
li_loop =  &
   dw_1.inv_filter.of_GetRegisterable  &
     (ls_columns)
lb_1.Reset()
IF li_loop <> 0 THEN
  FOR li_count = 1 to li_loop
    lb_1.AddItem(ls_columns[li_count])
  NEXT
ELSE
  lb_1.AddItem("No Registerable columns")
END IF
```

of_GetStyle

Description

Retrieves the current filter style. The filter style specifies which filter dialog box displays when you call the pfc_FilterDlg event.

Access

Public

Syntax

dwcontrol.instancename.**of_GetStyle** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)

Return value

Integer. Returns the current filter style:

- 0—PowerBuilder Filter dialog box (default)
- 1—PFC's w_filterextended dialog box
- 2—PFC's w_filtersimple dialog box

Examples

This example calls the of_GetStyle function:

```
Integer   li_filter_style
String    ls_filter_style[3] = &
   {"Default", "Wizard", "Simple"}

li_filter_style = &
   dw_emplist.inv_filter.of_GetStyle()
MessageBox("Filter Service", "Filter style is: " &
   + ls_filter_style[li_filter_style + 1] )
```

of_GetVisibleOnly

Description

Reports whether the filter service uses visible columns only or all columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetVisibleOnly ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)

Return value

Boolean. Returns TRUE if the filter service uses visible columns only and FALSE if it uses all columns.

Examples

This example calls the of_GetVisibleOnly function:

```
IF dw_emplist.inv_filter.of_GetVisibleOnly() THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      ("Visible columns only")
ELSE
   gnv_app.of_GetFrame().SetMicroHelp &
      ("All columns used for filter")
END IF
```

of_SetExclude

Description

Establishes a list of columns to be excluded from Filter dialog boxes.

Access

Public

Syntax

dwcontrol.instancename.**of_SetExclude** (*excludecols*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
excludecols	String array specifying columns to be excluded from filter dialog boxes

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetExclude function:

```
String    ls_exclude[ ]

ls_exclude[1] = "dept_id"
dw_emplist.inv_filter.of_SetExclude(ls_exclude)
```

of_SetFilter

Description

Specifies a DataWindow filter or displays a Filter dialog box.

Access

Public

Syntax

dwcontrol.instancename.**of_SetFilter** (*filter*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
filter	String containing the filter expression. If this argument is NULL, the function displays a filter dialog box to obtain the filter expression

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You control the type of filter dialog box in the of_SetStyle function.

Examples

This example calls the of_SetFilter function:

```
String    ls_null
Integer   li_return

SetNull(ls_null)
dw_emplist.inv_filter.of_SetStyle(1)
li_return = &
   dw_emplist.inv_filter.of_SetFilter(ls_null)
gnv_app.of_GetFrame().SetMicroHelp &
   ("Return code = " + String(li_return))
dw_emplist.inv_filter.of_Filter( )
```

of_SetStyle

Description

Specifies the filter dialog box type:

Default PowerBuilder Filter dialog box
PFC's w_filterextended dialog box
PFC's w_filtersimple dialog box

Access

Public

Syntax

dwcontrol.instancename.**of_SetStyle** (*filterstyle*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
filterstyle	Integer or n_cst_dwsrv_filter constant specifying the filter style: ♦ **0 or DEFAULT** PowerBuilder Filter dialog box (default) ♦ **1 or EXTENDED** The w_filterextended dialog box ♦ **2 or SIMPLE** The w_filtersimple dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to specify a filter style before calling the of_SetFilter and of_Filter functions or the pfc_FilterDlg event.

Examples

This example calls the of_SetStyle function to enable display of the w_filterextended dialog box:

```
String    ls_null
```

```
Integer   li_return

SetNull(ls_null)
dw_emplist.inv_filter.of_SetStyle  &
  (dw_emplist.inv_filter.EXTENDED)
li_return =
dw_emplist.inv_filter.of_SetFilter(ls_null)
gnv_app.of_GetFrame().SetMicroHelp("Return code = " &
   + String(li_return))
dw_emplist.inv_filter.of_Filter( )
```

of_SetVisibleOnly

Description

Controls whether filter dialog boxes display all columns or visible columns only.

Access

Public

Syntax

dwcontrol.instancename.**of_SetVisibleOnly** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
boolean	Boolean indicating whether filter dialog boxes display visible columns only (TRUE) or all columns (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetVisibleOnly function:

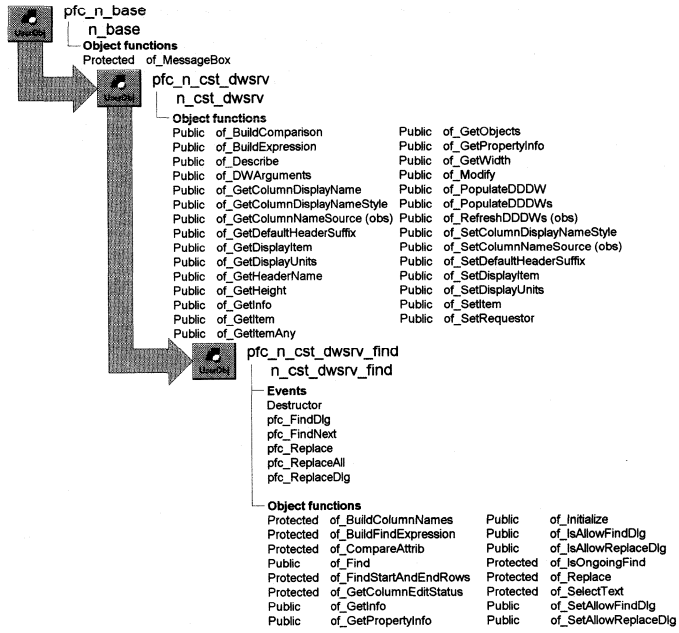
```
dw_emplist.inv_filter.of_SetVisibleOnly(TRUE)
```

n_cst_dwsrv_find

Description

Find and replace service. Provides functions and events that make it easy for you to add find and replace functionality to your application's DataWindows.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

Use this object to add find and replace capabilities to DataWindows.

To use this service:

1 Enable the service using the u_dw of_SetFind function:

```
dw_emp.of_SetFind(TRUE)
```

When users select Edit>Find or Edit>Replace from the menu bar of a menu that descends from the m_master menu, the service displays the w_find or w_replace dialog box. You can also display these dialog boxes programmatically, as explained in the next step.

2 Display the w_find dialog box by calling the u_dw pfc_FindDlg event; display the w_replace dialog box by calling the u_dw pfc_ReplaceDlg event. This example displays the w_find dialog box:

```
dw_emp.Event pfc_FindDlg()
```

See also

n_cst_dwsrv
n_cst_rtefind
u_dw

Instance variables

N_cst_dwsrv_find includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_allowfinddlg	Indicates whether the service displays the Find dialog	Boolean	Protected	Use of_SetAllowFindDlg and of_IsAllowFindDlg to access
ib_allowreplacedlg	Indicates whether the service displays the Replace dialog	Boolean	Protected	Use of_SetAllowReplaceDlg and of_IsAllowReplaceDlg to access
ib_ongoingfind	Indicates whether a find operation is in progress	Boolean	Protected	Internal
ib_replaced	Indicates whether a replace succeeded	Boolean	Protected	Internal
il_lastfindrow	Tracks the number of the last row found	Long	Protected	Internal
inv_findattrib	Structure object containing find and replace information	n_cst_findattrib	Protected	Internal

Instance variable	Description	Data type	Access	Usage
is_findexpression	Contains the string that is passed to the Find function	String	Protected	Internal
is_lastfindcolumn	Tracks the name of the last column found	String	Protected	Internal

Events

N_cst_dwsrv_find includes precoded events:

pfc_FindDlg	pfc_ReplaceAll
pfc_FindNext	pfc_ReplaceDlg
pfc_Replace	

pfc_FindDlg

Description

Displays the w_find dialog box.

Usage

This event is called when the DataWindow has focus and the user selects Edit>Find from a menu that descends from the PFC m_master menu.

pfc_FindNext

Description

Continues a search or finds the next occurrence.

Syntax

instancename.EVENT **pfc_FindNext** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_dwsrv_find
findattrib	n_cst_findattrib instance containing find information. From within the pfc_FindNext event, access this value through the *anv_findattrib* argument

Return value

Integer. Returns the number of the next matching row. This event returns 0 if no matching text is found and -1 if an error occurs.

Usage

This event is called by the w_find and w_replace dialog boxes to continue a search for a previously specified string.

pfc_Replace

Description

Replaces text as specified in the w_replace dialog box.

Syntax

instancename.Event **pfc_Replace** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_dwsrv_find
findattrib	n_cst_findattrib instance containing find information. From within the pfc_Replace event, access this value through the *anv_findattrib* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The w_replace dialog box calls this event when the user clicks Replace.

pfc_ReplaceAll

Description

Calls the pfc_Replace event to replace all occurrences of the search string with the target string.

Syntax

instancename.Event **pfc_ReplaceAll** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_dwsrv_find
findattrib	n_cst_findattrib instance containing find information. From within the pfc_ReplaceAll event, access this value through the *anv_findattrib* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The w_replace dialog box calls this event when the user clicks Replace All.

pfc_ReplaceDlg

Description

Displays the w_replace dialog box.

Usage

This event is called when the DataWindow has focus and the user selects Edit>Replace from a menu that descends from the PFC m_master menu.

Functions

N_cst_dwsrv_find includes precoded object functions:

of_BuildColumnNames	of_Initialize
of_BuildFindExpression	of_IsAllowFindDlg
of_CompareAttrib	of_IsAllowReplaceDlg
of_Find	of_IsOngoingFind
of_FindStartAndEndRows	of_Replace
of_GetColumnEditStatus	of_SelectText
of_GetInfo	of_SetAllowFindDlg
of_GetPropertyInfo	of_SetAllowReplaceDlg

of_BuildColumnNames

Description

Populates the inv_findattrib.is_lookdata and is_lookdisplay array elements with the columns to display in the w_find and w_replace dialog boxes.

Access

Protected

Syntax

dwcontrol.instancename.**of_BuildColumnNames** (*replacelist*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
replacelist	Boolean indicating whether to replace the list of columns flag

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_FindDlg event:

```
...
of_BuildColumnNames(FALSE)

//Open the Find dialog.
OpenWithParm(w_find, inv_findattrib)
...
```

of_BuildFindExpression

Description

Builds the string that is passed to the PowerScript Find function.

Access

Protected

Syntax

dwcontrol.instancename.**of_BuildFindExpression** (*find, column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
find	String being searched for
column	String specifying the column to search in

Return value

String. Returns the expression for use with the Find function if the function succeeds and ! if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_FindNext event:

```
...
//Build the Find expression.
is_findexpression = &
   of_BuildFindExpression(inv_findattrib.is_find, &
      inv_findattrib.is_lookdata &
         [inv_findattrib.ii_lookindex])
IF is_findexpression = '!' THEN Return 0
...
```

of_CompareAttrib

Description

Reports whether the user has made changes to specifications on the w_replace dialog box.

Access

Protected

Syntax

dwcontrol.instancename.**of_CompareAttrib** (*findattrib*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
findattrib	n_cst_findattrib instance containing find information

Return value

Boolean. Returns TRUE if there are no changes and FALSE if changes were made.

Usage

Internal.

Examples

This example is from the of_IsOnGoingFind function:

```
...
If IsValid(anv_findattrib) Then
   IF NOT of_CompareAttrib(anv_findattrib) THEN
      Return FALSE
   END IF
END IF

Return True
...
```

of_Find

Description

Searches for a value using the specified column, start row, and end row.

Access

Public

Syntax

dwcontrol.instancename.**of_Find** ({ *column, findstring, startrow, endrow* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
column (optional)	String specifying the column to be searched
findstring (optional)	String to be searched for
startrow (optional)	Long specifying the row at which to start the search

Argument	Description
endrow (optional)	Long specifying the row at which to end the search

Return value

Long. Returns the number of the first row containing *findstring* if the function succeeds, 0 if *findstring* is not found, or -1 if an error occurs.

Usage

Call this function to search for a string within a DataWindow. If you call this function with no arguments, PFC uses arguments from the previously specified of_Find function.

Examples

This example is from the of_Replace function:

```
...
CASE 'char'
   IF of_Find(as_colname, inv_findattrib.is_find, &
      al_row, al_row) = al_row THEN
         //Get the entire string.
         ls_find = idw_requestor.GetItemString &
            (al_row, as_colname)
      li_find_startpos = &
         Pos(Lower(ls_find), &
            Lower(inv_findattrib.is_find))
...
```

of_FindStartAndEndRows

Description

Determines the start and end row for a find.

Access

Protected

Syntax

dwcontrol.instancename.**of_FindStartAndEndRows** (*startrow*, *endrow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
startrow	Long into which the function places the start row (passed by reference)
endrow	Long into which the function places the end row (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, 0 if the last row is reached, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Find function:

```
...
ls_colname = &
   inv_findattrib.is_lookdata &
      [inv_findattrib.ii_lookindex]

//Determine the start and end rows for the find.
li_rc = of_FindStartAndEndRows &
      (ll_startrow, ll_endrow)
IF li_rc <= 0 THEN
   ib_ongoingfind = FALSE
   Return 0
END IF
...
```

of_GetColumnEditStatus

Description

Determines the edit status (editable, display-only, or protected) of the specified row/column location.

Access

Protected

Syntax

dwcontrol.instancename.**of_GetColumnEditStatus** (*column, row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
column	String specifying the column to test
row	String specifying the row to test

Return value

String. Returns values as follows:

- **Editable** The row/column is editable
- **Displayonly** The row/column is display-only
- **Protected** The row/column is not editable

- ! An error occurred

Usage

Internal.

Examples

This example is from the pfc_Replace event:

```
...
//Get the column edit status.
ls_coleditstatus = &
   of_GetColumnEditStatus(ls_colname, ll_currrow)

//Determine if the current row/col can be replaced.
IF ls_coleditstatus = 'editable' THEN
...
```

of_GetInfo

Description

Retrieves find service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_find.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the find service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
propertyobject	n_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_find.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_Initialize

Description

Initializes the inv_findattrib instance with the passed values.

Access

Public

Syntax

instancename.**of_Initialize** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_dwsrv_find
findattrib	n_cst_findattrib instance containing find information

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when the inv_findattrib default values are not appropriate.

Examples

This example calls the of_Initialize function:

```
// Display the Find/Replace windows with
// The Direction Control only.
n_cst_findattrib lnv_findattrib

lnv_findattrib.is_find = ''
lnv_findattrib.is_replacewith = ''
lnv_findattrib.ib_wholewordvisible =False
lnv_findattrib.ib_wholewordenabled =False
lnv_findattrib.ib_wholeword = False
lnv_findattrib.ib_matchcasevisible = False
lnv_findattrib.ib_matchcaseenabled = False
lnv_findattrib.ib_matchcase = False
lnv_findattrib.ib_directionvisible = True
lnv_findattrib.ib_directionenabled = True
lnv_findattrib.is_direction = 'Down'
lnv_findattrib.ib_lookvisible = False
lnv_findattrib.ib_lookenabled = False

dw_emp.inv_find.of_Initialize(lnv_findattrib)
```

of_IsAllowFindDlg

Description

Reports whether the Find dialog box is allowed.

Access

Public

Syntax

dwcontrol.instancename.**of_IsAllowFindDlg** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)

Return value

Boolean. Returns TRUE if the Find dialog box is allowed and FALSE if it is not.

Examples

This example calls the of_IsAllowFindDlg function:

```
IF NOT dw_1.inv_find.of_IsAllowFindDlg() THEN
   MessageBox("Find", "Find is not allowed")
ELSE
   MessageBox("Find", "Find is allowed")
END IF
```

of_IsAllowReplaceDlg

Description

Reports whether the Replace dialog box is allowed.

Access

Public

Syntax

dwcontrol.instancename.**of_IsAllowReplaceDlg** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)

Return value

Boolean. Returns TRUE if the Replace dialog box is allowed and FALSE if it is not.

Examples

This example calls the of_IsAllowReplaceDlg function:

```
IF NOT dw_1.inv_find.of_IsAllowReplaceDlg() THEN
   MessageBox("Replace", "Replace is not allowed")
ELSE
   MessageBox("Replace", "Replace is allowed")
END IF
```

of_IsOngoingFind

Description

Reports whether the current find operation is a new Find or Find Next operation.

Access

Protected

Syntax

dwcontrol.instancename.**of_IsOngoingFind** (*findattrib*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find

Argument	Description
findattrib	N_cst_findattrib instance containing find information

Return value

Boolean. Returns TRUE if the current find is the same as before and FALSE if it is not.

Usage

Internal.

Examples

This example is from the pfc_FindNext event:

```
String     ls_findexp
Long       ll_row

IF ib_ongoingfind THEN
//Determine if this continues as an ongoing Find.
   ib_ongoingfind = &
      of_IsOngoingFind(anv_findattrib)
END IF
...
```

of_Replace

Description

Replaces the specified row/column with the specified text.

Access

Protected

Syntax

*dwcontrol.instancename.***of_Replace** (*row, column, replacewith*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
row	Long specifying the row containing the column whose text is to be replaced
column	String specifying the column whose text is to be replaced
replacewith	String specifying the new text

Return value

Integer. Returns 1 if the function succeeds, -2 if *replacewith* is not a valid value for *column*, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_Replace event:

```
...
is_findexpression = &
    of_BuildFindExpression &
        (inv_findattrib.is_find, ls_colname)
IF is_findexpression = '!' THEN Return 0
IF idw_requestor.GetColumnName() = &
    ls_colname THEN
        // Replace the selected text.
        li_rc = of_Replace(ll_currrow, &
            ls_colname, &
                inv_findattrib.is_replacewith)
...
```

of_SelectText

Description

Selects the appropriate text at the specified row/column location. This function works for all column types, including dropdown DataWindows, dropdown listboxes, and edit masks.

Access

Protected

Syntax

dwcontrol.instancename.**of_SelectText** (*column*, *row*, *matchcase*, *find*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
column	String specifying the column name
row	Long specifying the row number
matchcase	Boolean indicating whether the search is case-sensitive
find	String specifying the find string

Return value

Integer. Returns 1 if the function succeeds and 0 if text selection is not appropriate.

Usage

Internal.

Examples

This example is from the of_Find function:

```
...
```

```
ls_coleditstatus = &
   of_GetColumnEditStatus(ls_colname, li_rowfound)
IF ls_coleditstatus = &
   'editable' OR ls_coleditstatus='readonly' THEN
      of_SelectText(ls_colname, li_rowfound, &
         inv_findattrib.ib_matchcase, &
            inv_findattrib.is_find)
END IF
...
```

of_SetAllowFindDlg

Description

Controls whether the Find dialog box is allowed.

Access

Public

Syntax

dwcontrol.instancename.**of_SetAllowFindDlg** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
boolean	Boolean specifying whether the Find dialog is allowed (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds -1 if an error occurs.

Examples

This example calls the of_SetAllowFindDlg function:

```
dw_1.inv_find.of_SetAllowFindDlg(FALSE)
```

of_SetAllowReplaceDlg

Description

Specifies whether the Replace dialog box is allowed.

Access

Public

Syntax

dwcontrol.instancename.**of_SetAllowReplaceDlg** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
boolean	Boolean specifying whether the Replace dialog is allowed (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds -1 if an error occurs.

Examples

This example calls the of_SetAllowReplaceDlg function:

```
dw_1.inv_find.of_SetAllowReplaceDlg(FALSE)
```

n_cst_dwsrv_linkage

Description

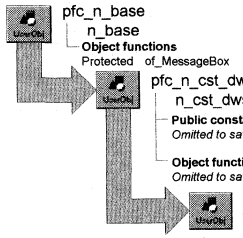
Service for linking DataWindow actions. This is typically used for Master/Detail processing.

N_cst_dwsrv_linkage contains the following features:

- **Linkage style** Controls whether detail DataWindows retrieve rows, filter rows, or scroll to the appropriate row
- **Update style** Controls how n_cst_dwsrv_linkage updates DataWindows (top-down, bottom-up, top-down then bottom-up, bottom-up then top-down, or a developer-specified custom update)
- **Custom update** Allows you to specify a customized update sequence
- **Delete style** When you delete a master row, this option specifies whether n_cst_dwsrv_linkage deletes detail rows, discards detail rows, or leaves them alone
- **Cascading key changes** N_cst_dwsrv_linkage automatically updates detail DataWindows when you change a key value on the master
- **Update on row change** When the master changes rows, this option updates the database with modifications made to detail DataWindows (retrieval style only)
- **Confirm on row change** When the master changes rows, this option displays a confirmation dialog box if modifications made to detail DataWindows will be lost (retrieval style only)
- **Confirm on delete** Displays a confirmation dialog box when the user deletes rows (retrieval style only)
- **Extended update** Allows you to integrate other controls (such as ListViews, TreeViews, and DataStores) into the default save process

N_cst_dwsrv_linkage is completed integrated with n_cst_luw and with the w_master pfc_Save process.

Ancestry



pfc_n_base
n_base
Object functions
Protected of_MessageBox

pfc_n_cst_dwsrv
n_cst_dwsrv
Public constants
Omitted to save space; use Browser to see all instance variables
Object functions
Omitted to save space; use Browser to see all functions

pfc_n_cst_dwsrv_linkage
n_cst_dwsrv_linkage
Events

Destructor	pfc_PreInsertRow
pfc_Clicked	pfc_RbuttonDown
pfc_DeleteRow	pfc_RetrieveEnd
pfc_InsertRow	pfc_RetrieveStart
pfc_ItemChanged	pfc_RowChanged
pfc_ItemFocusChanged	pfc_RowFocusChanged
pfc_PreDeleteRow	pfc_RowFocusChanging

Object functions

Public	of_AcceptText	Public	of_ResetArguments (obs)
Protected	of_CheckRequired	Public	of_ResetMaster
Public	of_ConfirmOnRowChange	Public	of_ResetUpdate
Public	of_CreateUpdateSequence	Public	of_Retrieve
Public	of_DeleteDetailRows	Protected	of_RetrieveDetails
Public	of_DeleteRows	Protected	of_Save
Protected	of_FilterDetails	Protected	of_ScrollDetails
Public	of_FindRoot	Public	of_SetArguments (obs)
Public	of_GetArguments (obs)	Public	of_SetConfirmOnDelete
Public	of_GetCustomUpdate	Public	of_SetConfirmOnRowChange
Public	of_GetDeleteStyle	Public	of_SetCustomUpdate
Public	of_GetDetails	Public	of_SetDeleteStyle
Public	of_GetInfo	Public	of_SetKeyCols
Public	of_GetMaster	Public	of_SetKeyValues
Public	of_GetOtherSaveObjects	Public	of_SetMaster
Public	of_GetPropertyInfo	Public	of_SetOtherSaveObjects
Public	of_GetRegistered	Public	of_SetRedraw
Public	of_GetSaveSound	Public	of_SetSaveSound
Public	of_GetStyle	Public	of_SetStyle
Protected	of_GetUpdateBottomUp (obs)	Public	of_SetSyncOnKeyChange
Public	of_GetUpdatesPending	Public	of_SetTransObject
Public	of_GetUpdateStyle	Public	of_SetUpdateBottomUp (obs)
Public	of_GetUseColLinks (obs)	Public	of_SetUpdateOnRowChange
Protected	of_GetValue (obs)	Public	of_SetUpdateRequestor
Public	of_IsConfirmOnDelete	Public	of_SetUpdateStyle
Public	of_IsConfirmOnRowChange	Public	of_SetUseColLinks (obs)
Public	of_IsKey	Public	of_TriggerEvent
Public	of_IsLinked	Public	of_Unlink (obs)
Public	of_IsRoot	Public	of_UnlinkDetail
Public	of_IsSyncOnKeyChange	Public	of_UnRegister
Public	of_IsUpdateOnRowChange	Public	of_Update
Public	of_ItemFocusChanged	Public	of_UpdateBottomUp
Public	of_KeyChanged	Public	of_UpdateBottomUpAndTopDown
Public	of_KeySync	Public	of_UpdateCustom
Public	of_LinkDetail	Public	of_UpdateOnRowChange
Public	of_LinkTo (obs)	Public	of_UpdatePrep
Public	of_PostUpdate	Public	of_UpdateTopDown
Public	of_Refresh	Public	of_UpdateTopDownAndBottomUp
Public	of_Register	Public	of_Validation
Public	of_Reset		

Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_linkageattrib
n_cst_platform
n_cst_propertyattrib
n_tr

Usage

Use this service to coordinate processing between related DataWindows.

You can specify the action taken by a detail DataWindow when row focus changes on the master. When row focus changes on the master DataWindow, the detail can:

Retrieve a new set of rows
Set a filter
Scroll to a new row

To use this service to control Master/Detail processing using columns in the master DataWindow as retrieval arguments for the detail DataWindow:

1 Enable the linkage service for the master DataWindow:

```
dw_master.of_SetLinkage(TRUE)
```

2 Enable the linkage service for the detail DataWindow:

```
dw_detail.of_SetLinkage(TRUE)
```

3 In the detail DataWindow, link to the Master:

```
dw_detail.inv_linkage.of_SetMaster(dw_master)
```

4 Register the related columns:

```
dw_detail.inv_linkage.of_Register &
  ("emp_id", "emp_id")
```

5 Establish the action taken in the detail when row focus changes in the master (this example retrieves a new set of rows):

```
dw_detail.inv_linkage.of_SetStyle &
  (dw_detail.inv_linkage.RETRIEVE)
```

6 Establish the transaction object for all DataWindows in the linkage chain by calling the linkage service's of_SetTransObject function for the top-level DataWindow:

```
dw_master.inv_linkage.of_SetTransObject(SQLCA)
```

7 Add retrieval logic to the DataWindows' pfc_Retrieve events (with the retrieval style, you only need to code pfc_Retrieve for the master DataWindow):

```
Return this.Retrieve()
```

8 Call the master DataWindow's of_Retrieve function:

```
IF dw_master.of_Retrieve() = -1 THEN
   MessageBox("Error", "Retrieval error")
ELSE
   dw_master.SetFocus()
```

```
END IF
```

9 Call other linkage service functions as needed.

Start from the top
When you want an action to ripple down through the chain of linked DataWindows, call the function in the topmost DataWindow.

See also

n_cst_luw
u_dw
w_master

Instance variables

N_cst_dwsrv_linkage includes instance variables for Internal:

Instance variable	Description	Data type	Access	Usage
BOTTOMUP	Constant set to 2	Integer	Public	Use with of_SetUpdateStyle
BOTTOMUP_TOPDOWN	Constant set to 4	Integer	Public	Use with of_SetUpdateStyle
CUSTOM	Constant set to 101	Integer	Public	Use with of_SetUpdateStyle
DELETE_ROWS	Constant set to 1	Integer	Public	Use with of_SetDeleteStyle
DISCARD_ROWS	Constant set to 2	Integer	Public	Use with of_SetDeleteStyle
FAILURE	Constant set to -1	Integer	Public	Internal
FILTER	Constant set to 1	Integer	Public	Use with of_SetStyle
ib_confirmondelete	Indicates whether the user is prompted to confirm deletions	Boolean	Protected	Set with of_SetConfirmOn Delete
ib_confirmonrowchange	Indicates whether the user is prompted to confirm modifications	Boolean	Protected	Set with of_SetConfirmOn RowChange

Instance variable	Description	Data type	Access	Usage
ib_retrieving	Indicates whether the DataWindow is retrieving data	Boolean	Protected	Internal
ib_synconkeychange	Indicates whether the linkage service cascades key changes to detail DataWindows	Boolean	Protected	Set with of_SetSyncOn KeyChange
ib_updatebottomup	Obsolete. Use ii_updatestyle	Boolean	Protected	Obsolete
ib_updateonrowchange	Indicates whether the service updates details when the master's row changes (retrieval style only)	Boolean	Protected	Set with of_SetUpdateOn RowChange
idw_details[ ]	Array of all DataWindows that are details to this DataWindow	u_dw	Protected	Internal
idw_master	Master to this DataWindow	u_dw	Protected	Internal
ii_collinkuse	Indicates how PFC uses column links to refresh data: ♦ 1 Filter ♦ 2 Retrieve ♦ 3 Scroll	Integer	Protected	Internal
ii_customupdate	Custom update indicator	Integer	Protected	Set with of_SetCustom Update
ii_deletestyle	Delete style indicator	Integer	Protected	Set with of_SetDeleteStyle
ii_updatestyle	Indicates how PFC performs updates	Integer	Protected	Internal
il_priorrow	Tracks rows	Long	Protected	Internal
inv_linkargs	Structure object containing linkage information	n_cst_ linkageattrib	Protected	Internal

Instance variable	Description	Data type	Access	Usage
inv_platform	Platform services	n_cst_platform	Protected	Internal
ipo_othersaveobjects[ ]	Additional controls to be updated when row focus changes	PowerObject	Protected	Set with of_SetOther SaveObjects
is_filterexp	Expression used to filter rows in detail DataWindows. This is used if ii_collinkuse is set to 1	String	Protected	Internal
is_keycols[ ]	Column that controls Master/Detail refresh processing. This is used if ii_collinkuse is set to 2	String	Protected	Internal
is_keycolsvalue[ ]	Saves keys	String	Protected	Internal
is_savesound	Sound file to be played for a successful save	String	Protected	Set with of_SetSaveSound
RETRIEVE	Constant set to 2	Integer	Public	Use with of_SetStyle
SCROLL	Constant set to 3	Integer	Public	Use with of_SetStyle
TOPDOWN	Constant set to 1	Integer	Public	Use with of_SetUpdateStyle
TOPDOWN_BOTTOMUP	Constant set to 3	Integer	Public	Use with of_SetUpdateStyle

Events

N_cst_dwsrv_linkage contains precoded user events:

Destructor	pfc_PreInsertRow
pfc_Clicked	pfc_RButtonDown
pfc_DeleteRow	pfc_RetrieveEnd
pfc_InsertRow	pfc_RetrieveStart
pfc_ItemChanged	pfc_RowChanged
pfc_ItemFocusChanged	pfc_RowFocusChanged
pfc_PreDeleteRow	pfc_RowFocusChanging

Destructor

Description

Destroys the platform service instance (inv_platform instance variable).

Usage

This event executes when the service is destroyed.

pfc_Clicked

Description

Determines if focus is going to another row and continues, halts processing, or updates rows as appropriate.

Syntax

dwcontrol.instancename.EVENT **pfc_Clicked** (*x, y, row, dwobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
x	Integer indicating the *x* position of the mouse click. From within the pfc_Clicked event, access this value through the *ai_xpos* argument
y	Integer indicating the *y* position of the mouse click. From within the pfc_Clicked event, access this value through the *ai_ypos* argument
row	Long indicating the clicked row. From within the pfc_Clicked event, access this value through the *al_row* argument
dwobject	DWObject variable indicating the clicked object. This argument is passed by reference. From within the pfc_Clicked event, access this value through the *adwo_obj* argument

Return value

Integer. Returns 1 to prevent focus from changing, 0 to continue processing, and -1 if an error occurs.

Usage

This event is called by the u_dw Clicked event.

pfc_DeleteRow

Description

Notifies detail DataWindows that a row has been deleted and there is a new current row.

Syntax

dwcontrol.instancename.EVENT **pfc_DeleteRow** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the row that was deleted. From within the pfc_DeleteRow event, access this value through the *al_row* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw pfc_DeleteRow event calls this event.

pfc_InsertRow

Description

Establishes key values for a newly inserted row.

Syntax

dwcontrol.instancename.Event **pfc_InsertRow** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the row that was inserted. From within the pfc_InsertRow event, access this value through the *al_row* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw pfc_InsertRow event calls this event.

pfc_ItemChanged

Description

Synchronizes key values in detail DataWindows (if key synchronization is enabled via the of_SetSyncOnKeyChange function).

Syntax

dwcontrol.instancename.EVENT **pfc_ItemChanged** (*row, dwobject, data*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long indicating the number of the row in which the ItemChanged event occurred. From within the pfc_ItemChanged event, access this variable through the *al_row* argument
dwobject	DWObject variable indicating the changed item. From within the pfc_ItemChanged event, access this value through the *adwo_obj* argument
data	String containing the new data for the item.

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw ItemChanged event calls this event.

pfc_ItemFocusChanged

Description

Calls the of_ItemFocusChanged function.

Syntax

dwcontrol.instancename.EVENT **pfc_ItemFocusChanged** (*row, dwobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long indicating the number of the row in which the ItemFocusChanged event occurred. From within the pfc_ItemFocusChanged event, access this variable through the *al_row* argument
dwobject	DWObject variable indicating the changed item. From within the pfc_ItemFocusChanged event, access this value through the *adwo_obj* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw ItemFocusChanged event calls this event.

pfc_PreDeleteRow

Description

Provides notification that a row may be deleted and prepares for possible deletion processing.

Syntax

dwcontrol.instancename.EVENT **pfc_PreDeleteRow** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the number of the row to be deleted

Return value

Integer. Returns 1 if the event succeeds, 0 if the delete should be prevented, and -1 if an error occurs.

Usage

The u_dw pfc_PreDeleteRow event calls this event.

pfc_PreInsertRow

Description

Prepares for insertion processing.

Syntax

dwcontrol.instancename.EVENT **pfc_PreInsertRow** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the event succeeds, 0 if the insert should be prevented, and -1 if an error occurs.

Usage

The u_dw pfc_PreInsertRow event calls this event.

pfc_RButtonDown

Description

Determines if focus is going to another row and updates rows as appropriate.

Syntax

dwcontrol.instancename.EVENT **pfc_RButtonDown** (*x*, *y*, *row*, *dwobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
x	Integer indicating the *x* position of the mouse click. From within the pfc_RButtonDown event, access this value through the *ai_xpos* argument
y	Integer indicating the *y* position of the mouse click. From within the pfc_RButtonDown event, access this value through the *ai_ypos* argument
row	Long indicating the number of the row in which the RButtonDown event occurred. From within the pfc_RButtonDown event, access this variable through the *al_row* argument
dwobject	DWObject variable indicating the clicked item. From within the pfc_RButtonDown event, access this value through the *adwo_obj* argument

Return value

Integer. Returns 1 to continue processing, 0 to prevent focus from changing, and -1 if an error occurs.

Usage

The u_dw RButtonDown event calls this event.

pfc_RetrieveEnd

Description

Resets the ib_retrieving instance variable.

Syntax

*dwcontrol.instancename.*EVENT **pfc_RetrieveEnd** (*rowcount*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
rowcount	Long specifying the number of rows retrieved

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw RetrieveEnd event calls this event.

pfc_RetrieveStart

Description

Sets the ib_retrieving instance variable.

Syntax

dwcontrol.instancename.EVENT **pfc_RetrieveStart** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw RetrieveStart event calls this event.

pfc_RowChanged

Description

Determines whether a row change requires a database update.

Syntax

dwcontrol.instancename.EVENT **pfc_RowChanged** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw RowChanged event calls this event.

When using the retrieval style and automatic updates, this event could be triggered without the benefit of a RowFocusChanging notification (for example, by a filter or sort action). With the Retrieve Style and ib_updateonrowchange = TRUE there is the possibility of data requiring updates before going to the new row. You should call of_Validation and of_Update to force these actions before performing the filter or sort.

pfc_RowFocusChanged

Description

Notifies each of the detail DataWindows attached to the requester that row focus has changed and that the contents of the new current row require the DataWindows to be refreshed based on values in the requester's current row.

Syntax

dwcontrol.instancename.Event **pfc_RowFocusChanged** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the new row. From within the pfc_RowFocusChanged event, access this value through the *al_row* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The u_dw RowFocusChanged event calls this event.

pfc_RowFocusChanging

Description

Updates the DataWindow before row focus changes, if enabled via the of_SetUpdateOnRowChange function.

Syntax

dwcontrol.instancename.EVENT **pfc_RowFocusChanging** (*previousrow*, *newrow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
previousrow	Long specifying the previous row. From within the pfc_RowFocusChanging event, access this value through the *al_prevrow* argument
newrow	Long specifying the new row. From within the pfc_RowFocusChanging event, access this value through the *al_newrow* argument

Return value

Integer. Returns 1 if the event succeeds, 0 to prevent row focus from changing, and -1 if an error occurs.

Usage

The u_dw RowFocusChanging event calls this event.

Functions

N_cst_dwsrv_linkage includes precoded object functions:

of_AcceptText	of_ResetArguments
of_CheckRequired	of_ResetMaster
of_ConfirmOnRowChange	of_ResetUpdate
of_CreateUpdateSequence	of_Retrieve
of_DeleteDetailRows	of_RetrieveDetails
of_DeleteRows	of_Save
of_FilterDetails	of_ScrollDetails
of_FindRoot	of_SetArguments
of_GetArguments	of_SetConfirmOnDelete
of_GetCustomUpdate	of_SetConfirmOnRowChange
of_GetDeleteStyle	of_SetCustomUpdate
of_GetDetails	of_SetDeleteStyle
of_GetInfo	of_SetKeyCols
of_GetMaster	of_SetKeyValues
of_GetOtherSaveObjects	of_SetMaster
of_GetPropertyInfo	of_SetOtherSaveObjects
of_GetRegistered	of_SetRedraw
of_GetSaveSound	of_SetSaveSound
of_GetStyle	of_SetStyle
of_GetUpdateBottomUp	of_SetSyncOnKeyChange
of_GetUpdatesPending	of_SetTransObject
of_GetUpdateStyle	of_SetUpdateBottomUp
of_GetUseColLinks	of_SetUpdateOnRowChange
of_GetValue	of_SetUpdateRequestor
of_IsConfirmOnDelete	of_SetUpdateStyle
of_IsConfirmOnRowChange	of_SetUseColLinks
of_IsKey	of_TriggerEvent
of_IsLinked	of_Unlink
of_IsRoot	of_UnlinkDetail
of_IsSyncOnKeyChange	of_UnRegister
of_IsUpdateOnRowChange	of_Update
of_ItemFocusChanged	of_UpdateBottomUp
of_KeyChanged	of_UpdateBottomUpAndTopDown
of_KeySync	of_UpdateCustom
of_LinkDetail	of_UpdateOnRowChange
of_LinkTo	of_UpdatePrep
of_PostUpdate	of_UpdateTopDown
of_Refresh	of_UpdateTopDownAndBottomUp
of_Register	of_Validation
of_Reset	

of_AcceptText

Description

Performs an AcceptText function for all DataWindows in the linkage chain, starting with the current DataWindow.

Access

Public

Syntax

*dwcontrol.instancename.***of_AcceptText** ({ *focusonerror* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
focusonerror (optional)	Boolean indicating whether PFC sets focus to the DataWindow when an error occurs (the default is FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

U_dw calls this function as part of the default save process. This function is part of the self-updating object API.

Examples

This example is from the u_dw the of_AcceptText function:

```
...
If IsValid(inv_linkage) Then
   li_rc = inv_linkage.of_AcceptText(ab_focusonerror)
Else
   li_rc = this.Event pfc_AcceptText(ab_focusonerror)
End If
...
```

of_CheckRequired

Description

Determines if any required columns contain NULL values in a specified row.

Access

Protected

Syntax

*dwcontrol.instancename.***of_CheckRequired** (*buffer, row, column, colname, updateonly*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
buffer	DWBuffer variable indicating the DataWindow buffer to check for required fields
row	Long specifying the number of the row to check. This argument also returns the number of the row in error (passed by reference)
column	Integer specifying the number of the first column to check. This argument also returns the number of the column in error (passed by reference)
colname	String into which the function places the name of the column in error (passed by reference)
updateonly	Boolean indicating whether you want to validate all rows and columns or only rows that have been inserted or modified

Return value

Integer. Returns 1 if there were columns with NULL values, 0 if there were no columns with NULL value, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Save function:

```
...
ll_checkrow = al_validaterow
   li_checkcolumn = 1
   li_rc = of_CheckRequired &
      (primary!, ll_checkrow, li_checkcolumn,  &
        ls_checkcolname, lb_updateonly)
...
```

of_ConfirmOnDelete

Description

Reports whether the linkage service displays a confirmation dialog box when rows are deleted.

Access

Public

Syntax

dwcontrol.instancename.**of_ConfirmOnDelete** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if confirm on delete is required and FALSE if it is not.

Usage

This function is for retrieval style only. You do not typically call it directly.

Examples

This example is from the of_Save function:

```
...
If ab_prompt Then
   If of_ConfirmOnDelete() Then
...
```

of_ConfirmOnRowChange

Description

Reports whether the linkage service displays a confirmation dialog box when a row change in a master DataWindow will causes updates to be lost in the detail DataWindow. This is accomplished by examining properties in the current DataWindow and the detail DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_ConfirmOnRowChange ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if confirm on row change is required and FALSE if it is not.

Usage

This function is for retrieval style only. You do not typically call it directly.

Examples

This example is from the of_Save function:

```
...
If ab_prompt Then
   If of_ConfirmOnRowChange() Then
...
```

of_CreateUpdateSequence

Description

Retrieves an array containing the default top-down update sequence.

Access

Public

Syntax

dwcontrol.instancename.**of_CreateUpdateSequence** (*sequence*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
sequence	U_dw array into which the function places references to linked DataWindows in the default update sequence (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

To determine the top-to-bottom sequence, access the array from the first element to the upper bound. To determine the bottom-up sequence, access the array from the upper bound down to one.

The linkage service uses this function to create a sequence for DataWindow updates.

Examples

This example calls the of_CreateUpdateSequence function:

```
Integer   li_i
Integer   li_max
u_dw      ldw_updatedw[ ]

// Create an array of the linked DataWindows
IF of_CreateUpdateSequence &
   (ldw_updatedw) <> 1 THEN Return -1

// Get the number of DataWindows
li_max = UpperBound(ldw_updatedw)

// Loop thru the array backwards to create
// a reverse update sequence.
FOR li_i = li_max to 1 STEP -1
   // Update the DataWindow
   IF ldw_updatedw[li_i].Event pfc_Update &
      (FALSE, FALSE) <> 1 THEN Return -1
```

```
NEXT

Return 1
```

of_DeleteDetailRows

Description

Deletes detail rows when the corresponding master row is deleted.

Access

Public

Syntax

dwcontrol.instancename.**of_DeleteDetailRows** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns the number of rows deleted or discarded if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_DeleteRow event:

```
...
of_SetRedraw(False)
li_rc = of_DeleteDetailRows()
this.Event pfc_RowFocusChanged(al_row)
...
```

of_DeleteRows

Description

Deletes or discards current DataWindows rows corresponding to the deleted row in the master.

Access

Public

Syntax

dwcontrol.instancename.**of_DeleteRows** (*delete, masterkeys, mastervalues*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
delete	Boolean whether to delete the rows (TRUE) or discard them (FALSE)
masterkeys	String array containing column names from the deleted master row
instancename	String array containing values from the deleted master row

Return value

Integer. Returns the number of rows deleted or discarded if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_DeleteDetailRows function:

```
...
CASE DELETE_ROWS
  li_rc =  &
   idw_details[li_detail].inv_Linkage.of_DeleteRows &
     (True, is_keycols, is_keycolsvalue)
CASE DISCARD_ROWS
  li_rc = &
    idw_details[li_detail].inv_Linkage.of_DeleteRows &
      (False, is_keycols, is_keycolsvalue)
...
```

of_FilterDetails

Description

Filters detail DataWindows based on current row values in the detail's master.

Access

Protected

Syntax

dwcontrol.instancename.**of_FilterDetails** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the current row of the master DataWindow control

Return value

Integer. Returns 1 if the function succeeds, 0 if refresh is not required, and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_FilterDetails function:

```
CHOOSE CASE of_GetStyle( )
   CASE FILTER
      Return of_FilterDetails(al_row)
   CASE RETRIEVE
      Return of_RetrieveDetails(al_row)
   CASE SCROLL
      Return of_ScrollDetails(al_row)
END CHOOSE
```

of_FindRoot

Description

Searches the linkage chain for the topmost DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_FindRoot** (*root*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
root	U_dw-based DataWindow into which the function places a reference to the topmost DataWindow (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function from any DataWindow in the linkage chain.

Examples

This example calls the of_FindRoot function:

```
...
IF of_FindRoot(ldw_root) <> 1 THEN Return -2
...
```

of_GetArguments

Description

Obsolete. Call of_GetRegistered instead.

Access

Public

Syntax

dwcontrol.instancename.**of_GetArguments** (*linkageinfo*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
linkageinfo	N_cst_linkageattrib structure object into which the function places linkage information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_GetCustomUpdate

Description

Retrieves the current custom update attribute.

Access

Public

Syntax

dwcontrol.instancename.**of_GetCustomUpdate** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns the current custom update attribute.

Usage

The linkage service architecture allows for multiple custom updates. Define an of_UpdateCustom function in the n_cst_dwsrv_linkage extension level object and call this function to determine the current custom update style.

Examples

This example calls the of_GetCustomUpdate function:

```
IF this.of_GetCustomUpdate() = 1 THEN
   Return this.of_UpdateBottomUp()
ELSE
   Return this.of_UpdateTopDown()
END IF
```

of_GetDeleteStyle

Description

Retrieves the current delete style.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDeleteStyle** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns the current delete style:

- **0 or DEFAULT** Default deletion style
- **1 or DELETE_ROWS** Delete rows from the database
- **2 or DISCARD_ROWS** Discard rows from the DataWindow buffer

Examples

This example calls the of_GetDeleteStyle function:

```
Integer  li_style

li_style =  &
   dw_orderitems.inv_linkage.of_GetDeleteStyle()
MessageBox("Delete Style", "Delete style is "  &
   + String(li_style))
```

of_GetDetails

Description

Creates an array containing references to all details for the current DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDetails** (*details*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (he u_dw default for this value is inv_linkage)
details	U_dw array into which the function places references to all details of *dwcontrol* (passed by reference)

Return value

Integer. Returns the number of elements in the *details* array.

Usage

Call this function from any DataWindow in the linkage chain.

Examples

This example calls the of_GetDetails function:

```
Integer   li_details
u_dw      ldw_details[ ]

li_details = &
   dw_emp.inv_linkage.of_GetDetails(idw_details)
...
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)
infoobject	n_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_linkage.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetMaster

Description

Retrieves a reference to the current DataWindows master.

Access

Public

Syntax

dwcontrol.instancename.**of_GetMaster** (*master*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
master	u_dw instance into which the function places a reference to the master DataWindow (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if there is no master reference.

Usage

All but the topmost DataWindow in the linkage chain have a master DataWindow.

Examples

This example calls the of_GetMaster function:

```
u_dw  ldw_master

IF  &
  dw_orderitems.inv_linkage.of_GetMaster(ldw_master) &
    = 1 THEN
      MessageBox("Master", "Master is "  &
        + ldw_master.ClassName())
END IF
```

of_GetOtherSaveObjects

Description

Retrieves the list of other objects, which are also updated when the UpdateOnRowChange functionality is enabled.

Access

Public

Syntax

dwcontrol.instancename.**of_GetOtherSaveObjects** (*objects*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
objects	PowerObject array into which the function places the list of other objects

Return value

Integer. Returns the number of objects in the objects *array* if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetOtherSaveObjects function:

```
PowerObject  lpo_obj[]
Integer  li_count

dw_1.inv_linkage.of_GetOtherSaveObjects(lpo_obj)
FOR li_count = 1 to UpperBound(lpo_obj)
  lb_1.AddItem(lpo_obj[li_count].ClassName())
NEXT
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)
propertyobject	n_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop
```

```
dw_1.inv_linkage.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegistered

Description

Retrieves the retrieval arguments that a master uses to populate a detail.

Access

Public

Syntax

*dwcontrol.instancename.***of_GetRegistered** (*linkattrib*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)
linkattrib	N_cst_linkageattrib instance into which the function places linkage argument values (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function from a detail DataWindow to access the retrieval arguments used by its master (set with of_Register).

Examples

This example calls the of_GetRegistered function:

```
n_cst_linkageattrib  lnv_link
Integer  li_count

dw_salesorder.inv_linkage.of_GetRegistered  &
  (lnv_link)
FOR li_count =  &
  1 TO UpperBound(lnv_link.is_mastercolarg)
    lb_1.AddItem(lnv_link.is_mastercolarg[li_count])
NEXT
FOR li_count =  &
  1 TO UpperBound(lnv_link.is_detailcolarg)
    lb_2.AddItem(lnv_link.is_detailcolarg[li_count])
NEXT
```

of_GetSaveSound

Description

Retrieves the name of the sound file, if enabled, that is played after a successful save.

Access

Public

Syntax

dwcontrol.instancename.**of_GetSaveSound** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

String. Returns the name of the sound file played after a successful save (if enabled via the of_SetSaveSound function).

Examples

This example calls the of_GetSaveSound function:

```
String  ls_sound
ls_sound = dw_cust.inv_linkage.of_GetSaveSound()
MessageBox("Sound", "Save sound is " + ls_sound)
```

of_GetStyle

Description

Retrieves the style for the current detail DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetStyle** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns the style:

- **1 or FILTER** Filter style
- **2 or RETRIEVE** Column retrieval style
- **3 or SCROLL** Scroll style

Examples

This example calls the of_GetStyle function:

```
Integer li_style

li_style = dw_orderitems.inv_linkage.of_GetStyle()
CHOOSE CASE li_style
CASE dw_orderitems.inv_linkage.FILTER
   MessageBox("Style", "Filter style")
CASE dw_orderitems.inv_linkage.RETRIEVE
   MessageBox("Style", "Retrieval style")
CASE dw_orderitems.inv_linkage.SCROLL
   MessageBox("Style", "Scroll style")
END CHOOSE
```

of_GetUpdateBottomUp

Description: Obsolete. Call of_GetUpdateStyle instead.

Access: Protected

Syntax: *dwcontrol.instancename*.**of_GetUpdateBottomUp ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value: Boolean. Returns TRUE if PFC performs updates from the bottom-up and FALSE if PFC performs updates from the top-down.

of_GetUpdatesPending

Description: Determines if there are updates pending in this DataWindow or any DataWindow below this one in the linkage chain.

Access: Public

Syntax: *dwcontrol.instancename*.**of_GetUpdatesPending ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns values as follows:

- 1 Updates are pending within the linkage chain
- 0 No updates pending within the linkage chain
- -1 An error occurred

Examples

This example calls the of_GetUpdatesPending function:

```
...
li_rc = ldw_root.inv_linkage.of_GetUpdatesPending( )
IF li_rc = -1 THEN Return -1
...
```

of_GetUpdateStyle

Description

Retrieves the update style for the current DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetUpdateStyle ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns the update style:

- **1 or TOPDOWN**
- **2 or BOTTOMUP**
- **3 or TOPDOWN_BOTTOMUP**
- **4 or BOTTOMUP_TOPDOWN**
- **101 or CUSTOM**

Examples

This example calls the of_GetUpdateStyle function:

```
Integer  li_count

li_count = dw_cust.inv_linkage.of_GetUpdateStyle()
MessageBox("Update Style",  &
  + "Update style is " + String(li_count))
```

of_GetUseColLinks

Description

Obsolete. Call of_GetStyle instead.

Access

Public

Syntax

dwcontrol.instancename.**of_GetUseColLinks** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns the retrieval style:

- **1 or FILTER** Filter style
- **2 or RETRIEVE** Column retrieval style
- **3 or SCROLL** Scroll style

of_GetValue

Description

Obsolete.

Access

Protected

Syntax

dwcontrol.instancename.**of_GetValue** (*row, column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	String specifying the row containing the item to be accessed
column	String specifying the column containing the item to be accessed

Return value

String. Returns the specified value in a format suitable for DataWindow expressions.

of_IsConfirmOnDelete

Description

Reports whether the linkage service displays a dialog box asking the user to confirm deletions.

Access

Public

Syntax

dwcontrol.instancename.**of_IsConfirmOnDelete** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if confirm on delete is enabled and FALSE if it is not.

Usage

This applies only if the retrieval style is retrieve and update on row change has been enabled.

Examples

This example calls the of_IsConfirmOnDelete function:

```
IF dw_1.inv_linkage.of_IsConfirmOnDelete() = TRUE THEN
  MessageBox("Delete", "Confirm on delete enabled")
ELSE
  MessageBox("Delete", "Confirm on delete disabled")
END IF
```

of_IsConfirmOnRowChange

Description

Reports whether the linkage service displays a dialog box asking the user to confirm row changes in the master that might cause updates to be lost in the details.

Access

Public

Syntax

dwcontrol.instancename.**of_IsConfirmOnRowChange** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if confirm on row changed is enabled and FALSE if it is not.

Usage

This applies only if the retrieval style is retrieve and update on row change has been enabled.

Examples

This example calls the of_IsConfirmOnRowChange function:

```
IF dw_1.inv_linkage.of_IsConfirmOnRowChange() =  &
  TRUE THEN
    MessageBox("Confirm on Row Change", "Enabled")
ELSE
  MessageBox("Confirm on Row Change", "Disabled")
END IF
```

of_IsKey

Description

Reports whether the specified column is a key column.

Access

Public

Syntax

dwcontrol.instancename.**of_IsKey** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
column	String specifying the column to be tested

Return value

Boolean. Returns TRUE if *column* is a key column and FALSE if it is not.

Usage

Specify key columns by calling the of_Register function.

Examples

This example calls the of_IsKey function:

```
IF dw_1.inv_linkage.of_IsKey("cust_id") THEN
  MessageBox("ID", "Cust_id is registered")
ELSE
  MessageBox("ID", "Cust_id is not registered")
END IF
```

of_IsLinked

Description

Reports whether a DataWindow is linked.

Access

Public

Syntax

dwcontrol.instancename.**of_IsLinked ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if *dwcontrol* is part of a linkage chain and FALSE if it is not.

Examples

This example calls the of_IsLinked function:

```
IF dw_emp.inv_linkage.of_IsLinked( ) THEN
   dw_emp.inv_linkage.of_Update( )
ELSE
   Return
END IF
```

of_IsRoot

Description

Reports whether the current DataWindow is the topmost in the linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_IsRoot ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if *dwcontrol* is at the top of the linkage chain and FALSE if it is not.

Examples

This example calls the of_IsRoot function:

```
Integer   li_i
Integer   li_numlinks
Integer   li_rc
String    ls_coldata
String    ls_filter

IF of_IsRoot() THEN Return -1
...
```

of_IsSyncOnKeyChange

Description

Reports whether the linkage service cascades key changes through detail DataWindows.

Access

Public

Syntax

dwcontrol.instancename.**of_IsSyncOnKeyChange ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if the linkage service cascades key changes through detail DataWindows and FALSE if it does not.

Examples

This example calls the of_IsSyncOnKeyChange function:

```
IF dw_1.inv_linkage.of_IsSyncOnKeyChange() THEN
  MessageBox("SyncOnKeyChange", "Enabled")
ELSE
  MessageBox("SyncOnKeyChange", "Disabled")
END IF
```

of_IsUpdateOnRowChange

Description

Reports whether the linkage service automatically updates modified detail rows when row focus changes in the master. Applies to retrieval style only.

Access

Public

Syntax

dwcontrol.instancename.**of_IsUpdateOnRowChange ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns TRUE if update on row change is enabled and FALSE if it is not.

Examples

This example calls the of_IsUpdateOnRowChange function:

```
IF dw_1.inv_linkage.of_IsUpdateOnRowChange() THEN
  MessageBox("UpdateOnRowChange", "Enabled")
ELSE
  MessageBox("UpdateOnRowChange", "Disabled")
END IF
```

of_ItemFocusChanged

Description

Tracks the row number and name of the column with focus.

Access

Public

Syntax

dwcontrol.instancename.**of_ItemFocusChanged** (*row*, *column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the row with focus
column	String specifying the column with focus

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The pfc_ItemFocusChanged event calls this function.

Examples

This example calls the of_ItemFocusChanged function:

```
...
of_ItemFocusChanged(al_row, adwo_object.Name)
```

of_KeyChanged

Description

Updates detail DataWindows with changes to the master's key.

Access

Public

Syntax

dwcontrol.instancename.**of_KeyChanged** (*row*, *column*, *prevvalue*, *newvalue*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the row number of the changed column in the master DataWindow
column	String specifying the key column in the master DataWindow
prevvalue	String containing the previous value
newvalue	Any containing the new value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Enable cascading key updates by calling of_SetSyncOnKeyChange.

Examples

This example is from the pfc_ItemChanged event:

```
...
of_KeyChanged(al_row, ls_colname, is_focusval,  &
   la_newvalue)
...
```

of_KeySync

Description

Synchronizes all detail DataWindows to the new key value.

Access

Public

Syntax

dwcontrol.instancename.**of_KeySync** (*row*, *column*, *prevvalue*, *newvalue*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the row number of the changed column in the master DataWindow
column	String specifying the key column in the master DataWindow
prevvalue	String containing the previous value
newvalue	Any containing the new value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The of_KeyChanged function calls this function.

Examples

This example is from the of_KeyChanged function:

```
...
li_rc = idw_details[li_i].inv_Linkage.of_KeySync &
  (al_row, as_column, as_prevvalue, aa_newvalue)
...
```

of_LinkDetail

Description

Adds the specified DataWindow to a master DataWindow's idw_details array.

Access

Public

Syntax

dwcontrol.instancename.**of_LinkDetail** (*detail*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
detail	U_dw-based DataWindow variable pointing to the detail DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The detail DataWindow's of_SetMaster function calls this function in the master DataWindow to establish the master/detail relationship.

Examples

This example calls the of_LinkDetail function:

```
IF IsValid(adw_master) THEN
   IF IsValid ( adw_master.inv_Linkage ) THEN
      IF adw_master.inv_Linkage.of_LinkDetail &
         (idw_Requestor) = 1 THEN
            idw_master = adw_master
            Return 1
      END IF
   END IF
END IF
```

of_LinkTo

Description

Obsolete. Call of_SetMaster instead.

Access

Public

Syntax

*dwcontrol.instancename.***of_LinkTo** (*master*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
master	DataWindow variable pointing to the master DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_PostUpdate

Description

Performs post update processing on all DataWindows in the linkage chain.

Access

Public

Syntax

*dwcontrol.instancename.***of_PostUpdate** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

U_dw calls this function as part of the default save process. This function is part of the self-updating object API.

Examples

This example is from the u_dw of_PostUpdate function:

```
...
If IsValid(inv_linkage) Then
   li_rc = inv_linkage.of_PostUpdate()
Else
   li_rc = this.Event pfc_postupdate()
End If
...
```

of_Refresh

Description

Refreshes detail DataWindows, based on the refresh type specified in the of_SetStyle function.

Access

Public

Syntax

dwcontrol.instancename.**of_Refresh** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the current row in the requesting DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The pfc_RowFocusChanged event calls this function when the current row changes in a master DataWindow. You do not typically call this function explicitly.

Examples

This example is from the pfc_RowFocusChanged event:

```
...
li_rc = &
   idw_details[li_detail].inv_Linkage.of_Refresh &
      (al_row)
...
```

of_Register

Description

Establishes the columns that relate master and detail DataWindows.

Access

Public

Syntax

dwcontrol.instancename.**of_Register** (*mastercol, detailcol*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
mastercol	String specifying the column in the master DataWindow that is a linkage argument

Argument	Description
detailcol	String specifying the column in the detail DataWindow that relates to *mastercol*

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function from the detail DataWindow.

Examples

This example calls the of_Register function:

```
dw_detail.inv_linkage.of_Register("id", "id")
```

of_Reset

Description

Clears data from the DataWindows in the linked chain, starting with the current DataWindow down through its linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_Reset ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage).

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Reset function:

```
dw_department.inv_linkage.of_Reset( )
```

of_ResetArguments

Description

Obsolete. Call of_UnRegister instead.

Access

Public

Syntax

dwcontrol.instancename.**of_ResetArguments ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_ResetMaster

Description

Unlinks the current detail DataWindow from its master.

Access

Public

Syntax

dwcontrol.instancename.**of_ResetMaster ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

After calling this function, the idw_master instance variable is NULL and should not be referenced.

Examples

This example calls the of_ResetMaster function:

```
dw_1.inv_linkage.of_ResetMaster()
```

of_ResetUpdate

Description

Clears the update flags for the DataWindows in the linked chain, starting with this DataWindow down through its linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_ResetUpdate ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to reset update flags after calling the of_Update function with the *resetflags* argument set to FALSE.

Examples

This example calls the of_ResetUpdate function if the of_Update function succeeds:

```
...
li_return = dw_master.inv_linkage.of_Update &
   (lb_accepttext, lb_reset_update)

IF li_return <> 1 THEN // Update failed
   ROLLBACK using SQLCA;
   IF SQLCA.SQLCODE <> 0 THEN
    MessageBox("RollBack Error", SQLCA.SQLErrText)
   ELSE
    MessageBox("Update Failed", "ROLLBACK Succeeded")
   END IF
ELSE
   COMMIT using SQLCA; // All updates succeeded
   IF SQLCA.SQLCODE <> 0 THEN
      MessageBox("Commit Error", SQLCA.SQLErrText)
      Return
   END IF
   dw_master.inv_linkage.of_ResetUpdate( )
END IF
```

of_Retrieve

Description

Retrieves the DataWindows in the linked chain, starting with this DataWindow down through the linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_Retrieve** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to coordinate retrieval among DataWindows in the linkage chain. You typically call this function from the uppermost DataWindow in the linkage chain. This function calls the DataWindow's pfc_Retrieve event.

For the retrieval style, you code a Retrieve function in the topmost DataWindow's pfc_Retrieve event. For the scroll and filter styles, you code a Retrieve function in each DataWindow's pfc_Retrieve event.

Examples

This example calls the of_Retrieve function:

```
dw_cust.of_SetLinkage(TRUE)
dw_salesorder.of_SetLinkage(TRUE)
dw_orderitems.of_SetLinkage(TRUE)

dw_salesorder.inv_linkage.of_SetMaster(dw_cust)
dw_salesorder.inv_linkage.of_Register("id", "cust_id")
dw_orderitems.inv_linkage.of_SetMaster(dw_salesorder)
dw_orderitems.inv_linkage.of_Register("id", "id")

dw_salesorder.inv_linkage.of_SetStyle  &
   (dw_salesorder.inv_linkage.RETRIEVE)
dw_orderitems.inv_linkage.of_SetStyle  &
   (dw_orderitems.inv_linkage.RETRIEVE)
dw_cust.inv_linkage.of_SetTransObject(SQLCA)

dw_cust.inv_linkage.of_Retrieve()
```

of_RetrieveDetails

Description

Retrieves rows for detail DataWindows based on the current row values in the detail's master.

Access

Protected

Syntax

dwcontrol.instancename.**of_RetrieveDetails** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the current row in the master DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Refresh function:

```
...
Choose Case of_GetStyle()
   Case FILTER
      Return of_FilterDetails(al_row)
   Case RETRIEVE
      Return of_RetrieveDetails(al_row)
   Case SCROLL
      Return of_ScrollDetails(al_row)
End Choose
...
```

of_Save

Description

Saves data for all DataWindows in the linkage chain as well as any other objects specified via the of_SetOtherSaveObjects function.

Access

Protected

Syntax

dwcontrol.instancename.**of_Save** (*validaterow*, *prompt*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
validaterow	Long specifying the row to validate. Specify 0 if there is no row to validate
prompt	Boolean specifying whether the user should be prompted to save changes (TRUE) or changes should be saved automatically (FALSE)

Return value

Integer. Returns values as follows:

- **2** Canceled
- **1** Successful
- **0** Updates pending
- **-1** An error occurred

Usage

This function saves data by calling the associated window's pfc_SaveObjects function. It is called by n_cst_dwsrv_linkage events when update on row change is enabled.

Examples

This example is from the pfc_RowChanged event:

```
...
If of_UpdateOnRowChange() Then
   li_rc = of_Save(0, False)
...
```

of_ScrollDetails

Description

Scrolls the DataWindows in the linked chain, starting with this DataWindow down through its linkage chain.

Access

Protected

Syntax

dwcontrol.instancename.**of_ScrollDetails** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the current row in the master DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Refresh function:

```
...
Choose Case of_GetStyle()
   Case FILTER
      Return of_FilterDetails(al_row)
   Case RETRIEVE
      Return of_RetrieveDetails(al_row)
   Case SCROLL
      Return of_ScrollDetails(al_row)
End Choose
...
```

of_SetArguments

Description

Obsolete. Call of_Register instead.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetArguments** (*mastercol, detailcol*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
mastercol	String specifying the column in the master DataWindow that is a linkage argument
detailcol	String specifying the column in the detail DataWindow that relates to *mastercol*

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_SetConfirmOnDelete

Description

Specifies whether the linkage service prompts users when deleting rows.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetConfirmOnDelete** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
boolean	Boolean specifying whether confirm on delete is enabled (TRUE) or disabled (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This specification only applies when using retrieval style and when you have enabled update on row change by calling of_SetUpdateOnRowChange(TRUE).

Examples

This example calls the of_SetConfirmOnDelete function:

```
dw_orditems.inv_linkage.of_SetStyle  &
```

```
    (dw_orditems.inv_linkage.RETRIEVE)
dw_orditems.inv_linkage.of_SetConfirmOnRowChange(TRUE)
dw_orditems.inv_linkage.of_SetConfirmOnDelete(TRUE)
dw_orditems.inv_linkage.of_SetUpdateOnRowChange(TRUE)
```

of_SetConfirmOnRowChange

Description

Specifies whether the linkage service prompts users when changing rows in the master will result in lost changes in this DataWindow. This specification affects row focus changing in the current DataWindow and all DataWindows in the linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_SetConfirmOnRowChange** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
boolean	Boolean specifying whether confirm on row change is enabled (TRUE) or disabled (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function in detail DataWindows. This specification only applies when using retrieval style and when you have enabled update on row change by calling of_SetUpdateOnRowChange(TRUE).

Examples

This example calls the of_SetConfirmOnRowChange function:

```
dw_orditems.inv_linkage.of_SetStyle  &
    (dw_orditems.inv_linkage.RETRIEVE)
dw_orditems.inv_linkage.of_SetConfirmOnRowChange(TRUE)
dw_orditems.inv_linkage.of_SetConfirmOnDelete(TRUE)
dw_orditems.inv_linkage.of_SetUpdateOnRowChange(TRUE)
```

of_SetCustomUpdate

Description

Specifies a custom update identifier.

Access

Public

Syntax

dwcontrol.instancename.**of_SetCustomUpdate** (*id*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
id	Integer from 1 to 9 specifying the custom update identifier

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when you need to establish more than one custom update sequence. Enable custom update by calling the of_SetUpdateStyle function.

Examples

This example calls the of_SetCustomUpdate function:

```
dw_cust.inv_linkage.of_SetCustomUpdate(9)
```

of_SetDeleteStyle

Description

Specifies the deletion style for this DataWindow's detail DataWindows. The deletion style controls whether the of_DeleteRows function leaves detail DataWindows alone, deletes detail rows from the database, or discards detail rows from the DataWindow buffer.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDeleteStyle** (*style*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
style	Integer specifying the delete style: ♦ **0 or DEFAULT** Detail rows remain when you delete a row in the master ♦ **1 or DELETE_ROWS** Deletes all rows of all corresponding detail DataWindows ♦ **2 or DISCARD_ROWS** Discards rows from the DataWindow buffer for all rows or all corresponding detail DataWindows

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function for master DataWindows to control how detail rows are affected when a master row is deleted.

Examples

This example calls the of_SetDeleteStyle function:

```
dw_cust.inv_linkage.of_SetDeleteStyle &
   (dw_cust.inv_linkage.DISCARD_ROWS)
```

of_SetKeyCols

Description

Establishes key columns on the master DataWindow. When the user changes data in a key column, the linkage service refreshes detail DataWindows automatically.

Access

Public

Syntax

dwcontrol.instancename.**of_SetKeyCols** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
column	String specifying the column in the master DataWindow that contains a linkage argument

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_SetKeyCols function:

```
...
Return idw_master.inv_Linkage.of_SetKeyCols &
   (as_masterarg)
```

of_SetKeyValues

Description

Retrieves key column values from the master DataWindow and sets the new value in detail DataWindow rows.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetKeyValues** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
row	Long specifying the row in the master DataWindow that contains the new value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_SetKeyValues function:

```
...
integer   li_rc

// Set the Key values for the new row.
li_rc = of_SetKeyValues(al_row)

Return li_rc...
```

of_SetMaster

Description

Links this DataWindow to the specified master DataWindow.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetMaster** (*masterdw*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
masterdw	U_dw-based DataWindow variable referencing the master DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMaster function:

```
dw_salesorder.inv_linkage.of_SetMaster(dw_cust)
```

```
dw_salesorder.inv_linkage.of_Register("id", "cust_id")
dw_orderitems.inv_linkage.of_SetMaster(dw_salesorder)
dw_orderitems.inv_linkage.of_Register("id", "id")
```

of_SetOtherSaveObjects

Description

Specifies other objects to be updated when update on row change is enabled.

Access

Public

Syntax

dwcontrol.instancename.**of_SetOtherSaveObjects** (*objects*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
objects	PowerObject array containing objects to be updated when row focus changes

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to enable DataWindows and other PowerBuilder controls to be updated when row focus changes. This specification only applies when using retrieval style and when you have enabled update on row change by calling of_SetUpdateOnRowChange(TRUE).

Examples

This example calls the of_SetOtherSaveObjects function:

```
PowerObject  lpo_objs[ ]

lpo_objs[1] = lv_salesinfo
dw_cust.inv_linkage.of_SetOtherSaveObjects(lpo_objs)
```

of_SetRedraw

Description

Sets redraw for this DataWindow and all detail DataWindows.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRedraw** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) redraw

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetRedraw function:

```
dw_cust.inv_linkage.of_SetRedraw(FALSE)
```

of_SetSaveSound

Description

Specifies the name of a sound file to play after a successful save.

Access

Public

Syntax

dwcontrol.instancename.**of_SetSaveSound** (*soundfile*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
soundfile	String specifying the name of the sound file to play after a successful save

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

To disable the save sound, set *soundfile* to an empty string. This feature is only used with the retrieval style.

Examples

This example calls the of_SetSaveSound function:

```
dw_cust.inv_linkage.of_SetSaveSound("slurp.wav")
```

of_SetStyle

Description

Specifies how the linkage service refreshes detail DataWindows:

- **Filter** The linkage service uses column links to build a filter and apply it to detail DataWindows

- **Retrieve** The linkage service uses column links as arguments to detail DataWindows
- **Scroll** The linkage service uses column links scroll detail DataWindows to the proper row

Access

Public

Syntax

dwcontrol.instancename.**of_SetStyle** (*style*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
style	Integer or n_cst_dwsrv_linkage constant specifying the style: ♦ **FILTER** ♦ **2 or RETRIEVE** ♦ **3 or SCROLL**

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Detail DataWindows call this function to specify how their rows are displayed.

If you will be updating rows through these DataWindows, it's best to use filter or scroll.

Examples

This example calls the of_SetStyle function:

```
dw_salesorder.inv_linkage.of_SetStyle &
   (dw_salesorder.inv_linkage.RETRIEVE)
dw_orderitems.inv_linkage.of_SetStyle &
   (dw_orderitems.inv_linkage.RETRIEVE)
```

of_SetSyncOnKeyChange

Description

Specifies whether the linkage service cascades key changes through detail DataWindows .

Access

Public

Syntax

dwcontrol.instancename.**of_SetSyncOnKeyChange** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
boolean	Boolean indicating whether key changes are cascaded through detail DataWindows (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function for every DataWindow in the linkage chain.

Examples

This example calls the of_SetSyncOnKeyChange function:

```
dw_cust.inv_linkage.of_SetSyncOnKeyChange(TRUE)
dw_salesorder.inv_linkage.of_SetSyncOnKeyChange(TRUE)
dw_orderitems.inv_linkage.of_SetSyncOnKeyChange(TRUE)
```

of_SetTransObject

Description

Sets the Transaction object for all DataWindows in the linked chain, from this DataWindow, on down through its linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_SetTransObject** (*transaction*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
transaction	N_tr variable specifying the Transaction object to use for the DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function on the top-level DataWindow to establish a Transaction object for DataWindows in the linkage chain.

Don't use the SetTransObject PowerScript function
Because this function calls the u_dw of_SetTransObject function, which establishes variables for use by other DataWindow services, call this function to specify a DataWindow's Transaction object instead of the SetTransObject PowerScript function.

Examples

This example calls the of_SetTransObject function. It assumes you have associated n_tr with SQLCA in the Application painter:

```
dw_master.inv_linkage.of_SetTransObject(SQLCA)
```

of_SetUpdateBottomUp

Description

Obsolete. Call of_SetUpdateStyle instead.

Access

Public

Syntax

dwcontrol.instancename.**of_SetUpdateBottomUp** (*direction*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
direction	Boolean indicating whether to update DataWindows from the bottom-up (TRUE) or from the top-down (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_SetUpdateOnRowChange

Description

Specifies whether the linkage service automatically updates modified detail rows when row focus changes in the master.

Access

Public

Syntax

dwcontrol.instancename.**of_SetUpdateOnRowChange** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function for detail DataWindows.

The update on row change feature is only enabled when using the retrieval style.

Examples

This example calls the of_SetUpdateOnRowChange function:

```
dw_salesorder.inv_linkage.of_SetUpdateOnRowChange &
  (TRUE)
dw_orderitems.inv_linkage.of_SetUpdateOnRowChange &
  (TRUE)
```

of_SetUpdateRequestor

Description

Calls the u_dw of_SetUpdateRequestor function for each DataWindow in the linkage chain.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetUpdateRequestor** (*requestor*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
requestor	PowerObject containing the object requesting the update

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SetUpdateRequestor function:

```
...
If IsNull(idw_details[li_i].inv_Linkage) Or &
   Not IsValid ( idw_details[li_i].inv_Linkage Then
 Return -1
END IF
```

```
li_rc =  &
 idw_details[li_i].inv_Linkage.of_SetUpdateRequestor &
   (apo_updaterequestor)
...
```

of_SetUpdateStyle

Description

Specifies the update style.

Access

Public

Syntax

dwcontrol.instancename.**of_SetUpdateStyle** (*updatestyle*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
updatestyle	Integer or n_cst_dwsrv_linkage constant specifying the update style: ◆ **1 or TOPDOWN** ◆ **2 or BOTTOMUP** ◆ **3 or TOPDOWN_BOTTOMUP** ◆ **4 or BOTTOMUP_TOPDOWN** ◆ **101 or CUSTOM**

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The update style controls the order in which the linkage service updates DataWindows in the linkage chain.

If you specify a custom update style, you must also create and code an of_UpdateCustom function in the n_cst_dwsrv_linkage extension-level object. This function then updates the DataWindows in the desired order.

Examples

This example calls the of_SetUpdateStyle function:

```
dw_cust.inv_linkage.of_SetUpdateStyle  &
   (dw_cust.inv_linkage.TOPDOWN)
```

of_SetUseColLinks

Description

Obsolete. Call of_SetStyle instead.

Access

Public

Syntax

dwcontrol.instancename.**of_SetUseColLinks** (*linktype*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
linktype	Integer specifying the linkage type: ♦ 1 PFC uses columns in the master DataWindow as filter criteria for detail DataWindows ♦ 2 PFC uses columns in the master DataWindow as retrieval arguments for the detail DataWindow ♦ 3 PFC uses columns in the master DataWindow to control scrolling in the detail DataWindow

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_TriggerEvent

Description

Triggers the specified user event in the DataWindows in the linked chain, starting with this DataWindow down through its linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_TriggerEvent** (*userevent*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
userevent	String containing the name of a valid user event for the DataWindow control

Return value

Integer. Returns 1 if the function succeeds, -1 if *userevent* did not exist (or had no code) in at least one DataWindow, and -2 if an error occurs.

Examples

This example calls the of_TriggerEvent function:

```
dw_master.inv_linkage.of_TriggerEvent  &
   ("ue_custom_update")
```

of_Unlink

Description

Obsolete. Call of_ResetMaster instead.

Access

Public

Syntax

dwcontrol.instancename.**of_Unlink** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_UnlinkDetail

Description

Removes the specified DataWindow from a master DataWindow's idw_details array.

Access

Public

Syntax

dwcontrol.instancename.**of_UnlinkDetail** (*detaildatawindow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
detaildatawindow	U_dw DataWindow variable pointing to the detail DataWindow to be unlinked

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_ResetMaster function:

```
...
If idw_master.inv_Linkage.of_UnlinkDetail  &
```

```
    (idw_Requestor) <> 1 Then
      Return -1
End If
...
```

of_UnRegister

Description

Removes the current DataWindow from the linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_UnRegister** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You cannot call this function on the topmost DataWindow in the linkage chain.

Examples

This example calls the of_UnRegister function:

```
dw_orderitems.inv_linkage.of_UnRegister()
```

of_Update

Description

Updates the linked DataWindows in the direction specified in the of_SetUpdateStyle function. Depending on the arguments specified, this function calls the PowerScript AcceptText function before calling the Update function. You can also specify whether to clear each DataWindow's update flags.

Access

Public

Syntax

dwcontrol.instancename.**of_Update** ({ *accepttext* {, *resetflags* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Argument	Description
accepttext (optional)	Boolean indicating whether the DataWindow control should automatically perform an AcceptText before performing the update: TRUE—Perform AcceptText (default) FALSE—Do not perform AcceptText
resetflags (optional)	Boolean indicating whether *dwcontrol* should automatically reset the update flags: TRUE—Reset the flags (default) FALSE—Do not reset the flags

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

U_dw calls this function as part of the default save process. This function is part of the self-updating object API.

If *resetflags* is FALSE, the function does not reset the DataWindows' update flags. If the function succeeds, you should reset these flags by calling the of_ResetUpdate function.

Examples

This example is from the u_dw of_Update function:

```
...
If IsValid(inv_linkage) Then
   li_rc = inv_linkage.of_Update  &
      (ab_accepttext, ab_resetflag)
Else
   li_rc = this.Event pfc_Update  &
      (ab_accepttext, ab_resetflag)
End If
...
```

of_UpdateBottomUp

Description

Updates the DataWindow in the linked chain, starting with the current DataWindow and going up the chain.

Access

Public

Syntax

dwcontrol.instancename.**of_UpdateBottomUp** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function is called by the of_Update function when you have previously specified bottom-up update direction by calling of_SetUpdateStyle(BOTTOMUP). You don't typically call this function explicitly.

Examples

This example is from the of_Update function:

```
...
CHOOSE CASE ii_updatestyle
CASE TOPDOWN
   li_rc = of_UpdateTopDown()
CASE BOTTOMUP
   li_rc = of_UpdateBottomUp()
CASE TOPDOWN_BOTTOMUP
...
```

of_UpdateBottomUpAndTopDown

Description

Updates the DataWindow in the linked chain, starting with the current DataWindow and going up the chain (performing deletes) and then starting with the top DataWindow and going down the chain (performing inserts and updates).

Access

Public

Syntax

dwcontrol.instancename.**of_UpdateBottomUpAndTopDown** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function is called by the of_Update function when you have previously specified bottom-up/top-down update direction by calling of_SetUpdateStyle(BOTTOMUP_TOPDOWN). You don't typically call this function explicitly.

Examples

This example is from the of_Update function:

```
...
CHOOSE CASE ii_updatestyle
CASE TOPDOWN
   li_rc = of_UpdateTopDown()
CASE BOTTOMUP
   li_rc = of_UpdateBottomUp()
CASE TOPDOWN_BOTTOMUP
   li_rc = of_UpdateTopDownAndBottomUp()
CASE BOTTOMUP_TOPDOWN
   li_rc = of_UpdateBottomUpAndTopDown()
CASE CUSTOM
...
```

of_UpdateCustom

Description

Empty user event that contains custom update processing.

Access

Public

Syntax

dwcontrol.instancename.**of_UpdateCustom** (*id*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
id	Integer identifying the custom update, as specified in a previously called of_SetCustomUpdate function

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The of_Update function calls this function when you set the update style to Custom via the of_SetUpdateStyle function.

You must implement an of_UpdateCustom function in the n_cst_dwsrv_linkage extension level object. This function checks the passed argument (if there are multiple custom update possibilities) and updates DataWindows in the appropriate order.

Examples

This example is from the of_Update function:

```
...
CHOOSE CASE ii_updatestyle
CASE TOPDOWN
   li_rc = of_UpdateTopDown()
CASE BOTTOMUP
   li_rc = of_UpdateBottomUp()
CASE TOPDOWN_BOTTOMUP
   li_rc = of_UpdateTopDownAndBottomUp()
CASE BOTTOMUP_TOPDOWN
   li_rc = of_UpdateBottomUpAndTopDown()
CASE CUSTOM
   li_rc = of_UpdateCustom(ii_customupdate)
...
```

of_UpdateOnRowChange

Description

Polls the current DataWindow and all detail DataWindows to see whether any updates are required before changing rows.

Access

Public

Syntax

dwcontrol.instancename.**of_UpdateOnRowChange ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Boolean. Returns TRUE if updates are required and FALSE if they are not.

Usage

Internal.

Examples

This example is from the pfc_DeleteRow event:

```
...
If of_UpdateOnRowChange() Then
   li_rc = of_Save(0, FALSE)
```

```
...
```

of_UpdatePrep

Description

Prepares all linked DataWindows for update.

Access

Public

Syntax

dwcontrol.instancename.**of_UpdatePrep** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

U_dw calls this function as part of the default save process. This function is part of the self-updating object API.

Examples

This example is from the u_dw of_UpdatePrep function:

```
...
If IsValid(inv_linkage) Then
   li_rc = inv_linkage.of_UpdatePrep()
Else
   li_rc = this.Event pfc_UpdatePrep()
End If
...
```

of_UpdateTopDown

Description

Updates the DataWindow in the linked chain, starting with the current DataWindow and going down the chain.

Access

Public

Syntax

dwcontrol.instancename.**of_UpdateTopDown** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function is called by the of_Update function when you have previously specified top-down update direction by calling of_SetUpdateStyle(TOPDOWN). You don't typically call this function explicitly.

Examples

This example is from the of_Update function:

```
...
CHOOSE CASE ii_updatestyle
CASE TOPDOWN
   li_rc = of_UpdateTopDown()
CASE BOTTOMUP
   li_rc = of_UpdateBottomUp()
CASE TOPDOWN_BOTTOMUP
...
```

of_UpdateTopDownAndBottomUp

Description

Updates the DataWindows in the linked chain, starting with the top DataWindow and going down the chain (performing inserts and updates) and then starting with the bottom DataWindow and going up the chain (performing deletes).

Access

Public

Syntax

dwcontrol.instancename.**of_UpdateTopDownAndBottomUp** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function is called by the of_Update function when you have previously specified top-down/bottom-up update direction by calling of_SetUpdateStyle(TOPDOWN_BOTTOMUP). You don't typically call this function explicitly.

Examples

This example is from the of_Update function:

```
...
```

```
CASE TOPDOWN
   li_rc = of_UpdateTopDown()
CASE BOTTOMUP
   li_rc = of_UpdateBottomUp()
CASE TOPDOWN_BOTTOMUP
   li_rc = of_UpdateTopDownAndBottomUp()
CASE BOTTOMUP_TOPDOWN
   li_rc = of_UpdateBottomUpAndTopDown()
...
```

of_Validation

Description

Performs validation on all DataWindows in the linkage chain.

Access

Public

Syntax

dwcontrol.instancename.**of_Validation ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

Return value

Integer. Returns 1 if the function succeeds and -1 if a validation error occurs.

Usage

U_dw calls this function as part of the default save process. This function is part of the self-updating object API.

Examples

This example is from the u_dw of_Validation function:

```
...
If IsValid(inv_linkage) Then
   li_rc = inv_linkage.of_Validation()
Else
   li_rc = this.Event pfc_Validation()
End If
...
```

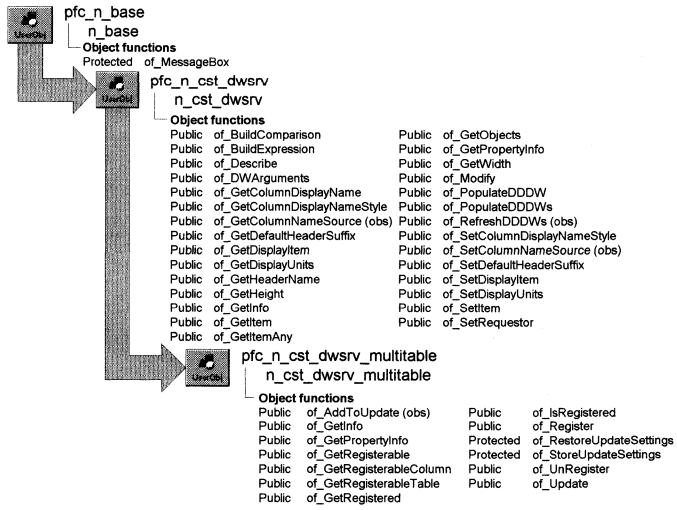
n_cst_dwsrv_multitable

Description

A collection of functions that facilitate update operations for DataWindows that contain rows from more than one table.

N_cst_dssrv_multitable is the same as n_cst_dwsrv_multitable
N_cst_dssrv_multitable, the DataStore multitable update service object, is almost identical to n_cst_dwsrv_multitable except that it applies to DataStores instead of DataWindow controls. Except where noted, references to n_cst_dwsrv_multitable in the following discussions apply equally to n_cst_dssrv_multitable.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

Use this object to handle update operations for multiple tables within a DataWindow.

To use this service:

1 Enable the service using the u_dw of_SetMultiTable function:

```
dw_project.of_SetMultiTable(TRUE)
```

2 Call the of_Register function once for each table to be updated in a multitable update:

```
String  ls_projcols[ ] = {"project_proj_id"}
String  ls_taskcols[ ] =  &
   {"task_proj_id", "task_task_id"}

dw_project.inv_multitable.of_Register &
   ("project", ls_projcols)
dw_project.inv_multitable.of_Register &
   ("task", ls_taskcols)
```

3 Update all specified tables by calling the w_master pfc_Save event:

```
Integer   li_return

li_return = w_projlist.Event pfc_Save( )
```

Structures

Structure	Field	Data type
os_updcharacteristics	Boolean	b_keyupdateinplace
	Integer	i_whereoption
	String	s_keycolumns[ ]
	String	s_updatecolumns[ ]
	String	s_updatetable

See also

n_cst_dwsrv_rowmanager
n_cst_dwsrv_linkage
u_dw

Instance variables

N_cst_dwsrv_multitable includes one instance variable:

Instance variable	Description	Data type	Access	Usage
istr_newupdate[ ]	Used to control multiple updates	os_updcharacteristics	Protected	Internal

Functions

N_cst_dwsrv_multitable includes precoded object functions:

of_AddToUpdate	of_IsRegistered
of_GetInfo	of_Register
of_GetPropertyInfo	of_RestoreUpdateSettings
of_GetRegisterable	of_StoreUpdateSettings
of_GetRegisterableColumn	of_UnRegister
of_GetRegisterableTable	of_Update
of_GetRegistered	

of_AddToUpdate

Description

Obsolete. Call of_Register instead.

Access

Public

Syntax

dwcontrol.instancename.**of_AddToUpdate** (*table*, *keycolumns* {, *updateablecols* {, *keyinplace*, *whereoption* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
table	String specifying the table name
keycolumns	String array containing the key columns used for update
updateablecols (optional)	String array containing the list of updatable columns (the default is all columns in the DataWindow)
keyinplace (optional)	Boolean indicating whether to update the key in place (TRUE) or delete and reinsert (FALSE) (the default is FALSE). If you specify this argument, you must also specify *whereoption*
whereoption (optional)	Integer specifying the DataWindow Update Where option: ♦ 0 Key columns ♦ 1 Key and updatable columns ♦ 2 Key and modified columns If you specify this argument, you must also specify *keyinplace*

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_multitable.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information:

Not in n_cst_dssrv_multitable
This function is not implemented in n_cst_dssrv_multitable.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_multitable.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegisterable

Description

Retrieves an array of columns in the current DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterable** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
columns	String array into which the function places a list of columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Usage

The DataWindow Properties window calls this function.

Examples

This example calls the of_GetRegisterable function:

```
String   ls_columns[ ]
Integer  li_count, li_loop
li_loop =  &
```

```
   dw_1.inv_multitable.of_GetRegisterable  &
     (ls_columns)
lb_1.Reset()
IF li_loop <> 0 THEN
  FOR li_count = 1 to li_loop
    lb_1.AddItem(ls_columns[li_count])
  NEXT
ELSE
  lb_1.AddItem("No Registerable columns")
END IF
```

of_GetRegisterableColumn

Description

Retrieves all registerable columns for the passed table.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterableColumn** (*table*, *columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
table	String specifying the name of the table
columns	String array into which the function places all registerable columns for *table* (passed by reference)

Return value

Integer. Returns the number of elements in the *columns* array if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetRegisterableColumn function:

```
String  ls_cols[ ]
Integer  li_count, li_loop

li_count =  &
  dw_1.inv_multitable.of_GetRegisterableColumn  &
    ("employee", ls_cols)
lb_1.Reset()
FOR li_loop = 1 to li_count
   lb_1.AddItem(ls_cols[li_loop])
NEXT
```

of_GetRegisterableTable

Description

Retrieves an array of registerable tables whose columns are used in the current DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterableTable** (*tables*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
tables	String array into which the function places the names of tables whose columns are used in the current DataWindow (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function.

Examples

This example calls the of_GetRegisterableTable function:

```
String  ls_tables[ ]
Integer  li_count, li_loop

li_count =  &
  dw_1.inv_multitable.of_GetRegisterableTable  &
   (ls_tables)
lb_1.Reset()
FOR li_loop = 1 to li_count
   lb_1.AddItem(ls_tables[li_loop])
NEXT
```

of_GetRegistered

Description

Retrieves information on registered columns for a specified table.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegistered** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
columns	String array into which the function places a list of registered columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Examples

This example calls the of_GetRegistered function:

```
String  ls_empkey[ ]
String  ls_updatable[ ]
Boolean  lb_keyinplace
Integer  li_whereoption
Integer  li_count, li_loop

li_count =  &
  dw_1.inv_multitable.of_GetRegistered  &
    ("employee", ls_empkey, ls_updatable,  &
      lb_keyinplace, li_whereoption)
lb_1.Reset()
FOR li_loop = 1 to li_count
   lb_1.AddItem(ls_empkey[li_loop])
NEXT
```

of_IsRegistered

Description

Reports whether a table is registered.

Access

Public

Syntax

dwcontrol.instancename.**of_IsRegistered** (*table*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
table	String specifying the table to be checked

Return value

Boolean. Returns TRUE if *table* is registered and FALSE if it is not.

Examples

This example calls the of_IsRegistered function:

```
IF NOT dw_1.inv_multitable.of_IsRegistered  &
  ("department") THEN
    Return -1
END IF
```

of_Register

Description

Registers a table (and its corresponding update characteristics) with the multitable update service.

Access

Public

Syntax

dwcontrol.instancename.**of_Register** (*table*, *keycolumns* { *updatablecolumns* {, *keyinplace*, *whereoption* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
table	String specifying the table to be registered
keycolumns	String array specifying the key columns for *table*
updatablecolumns (optional)	String array specifying updatable columns for *table*. The default is all columns
keyinplace (optional)	Boolean specifying whether to update the key in place (TRUE) or delete then insert if the key changes (FALSE) (the default is FALSE). If you specify this argument, you must also specify *whereoption*
whereoption (optional)	Integer specifying the WHERE option: ◆ **0** Key columns only ◆ **1** Key columns and all updatable columns (default) ◆ **2** Key and modified columns If you specify this argument, you must also specify *keyinplace*

Return value

Integer. Returns the 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function after enabling the multitable update service.

Examples

This example calls the of_Register function:

```
String  ls_empcols[ ] = {"employee_emp_id"}
String  ls_deptcols[ ] = {"department_dept_id"}

dw_1.of_SetMultiTable(TRUE)
dw_1.inv_multitable.of_Register &
   ("employee", ls_empcols)
dw_1.inv_multitable.of_Register &
   ("department", ls_deptcols)
```

of_RestoreUpdateSettings

Description: Restores DataWindow update characteristics.

Access: Protected

Syntax: *dwcontrol.instancename*.**of_RestoreUpdateSettings** (*settings*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
settings	Os_updcharacteristics structure containing the update characteristics

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: Internal.

Examples: This example calls the of_RestoreUpdateSettings function:

```
...
li_rc = &
   of_RestoreUpdateSettings(lstr_originalupdate)
IF li_rc < 0 THEN Return li_rc
...
```

of_StoreUpdateSettings

Description: Stores DataWindow update characteristics.

Access: Protected

Syntax: *dwcontrol.instancename*.**of_StoreUpdateSettings** (*settings*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
settings	Os_updcharacteristics structure containing the update characteristics

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_StoreUpdateSettings function:

```
...
li_rc = &
   of_StoreUpdateSettings(lstr_originalupdate)
IF li_rc < 0 THEN Return li_rc
...
```

of_UnRegister

Description

Removes one or all tables from the list of tables handled by the multitable update service.

Access

Public

Syntax

dwcontrol.instancename.**of_UnRegister** ({ *table* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
table (optional)	String specifying the name of the table to be unregistered. If you omit this argument, the function unregisters all tables

Return value

Integer. Returns the number of tables unregistered if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_UnRegister function:

```
dw_1.inv_multitable.of_UnRegister  &
  ("department")
```

of_Update

Description

Saves rows for all registered tables in the DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_Update** ({ *accept* {, *resetflag* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
accept (optional)	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database. The default is TRUE
resetflag (optional)	Boolean indicating whether the Update function resets the update flags. The default is TRUE

Return value

Integer. Returns 1 if the function succeeds, -1 if an update error occurs, and -2 for all other errors.

Usage

Call this function to update all tables registered with the service.

Examples

This example is from the u_dw pfc_Update event:

```
...
IF IsValid (inv_multitable) THEN
   li_rc = inv_multitable.of_update  &
      (ab_accepttext, ab_resetflag)
ELSE
   li_rc = this.Update(ab_accepttext, ab_resetflag)
END IF

...
```

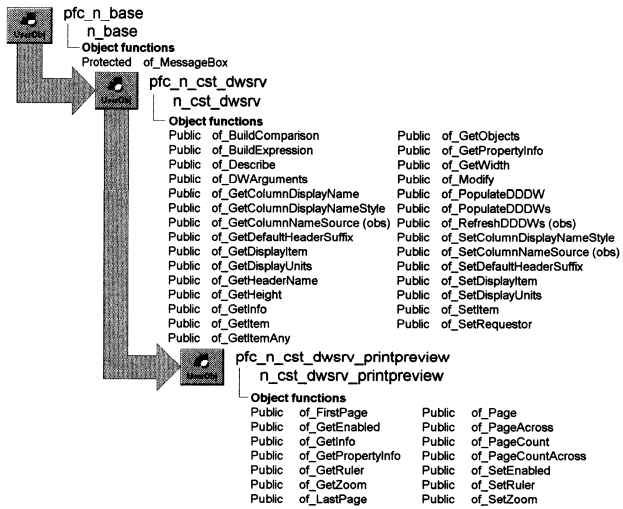
n_cst_dwsrv_printpreview

Description

Contains functions that allow you to control print preview. When this service is enabled, menus that descend from PFC's m_master menu have automatic access to print preview functionality (such as zoom, first page, last page, previous page, and next page).

N_cst_dssrv_printpreview is the same as n_cst_dwsrv_printpreview
N_cst_dssrv_printpreview, the DataStore print preview service object is identical to n_cst_dwsrv_printpreview except that it applies to DataStores instead of DataWindow controls. Except where noted, references to n_cst_dwsrv_printpreview in the following discussions apply equally to n_cst_dssrv_printpreview.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib
n_cst_string

Usage

Use this object to provide DataWindow print preview functionality.

To use this service:

1 Enable the service using the u_dw of_SetPrintPreview function:

```
dw_emp.of_SetPrintPreview(TRUE)
```

2 Toggle print preview mode by calling the u_dw pfc_PrintPreview event:

```
dw_emp.Event pfc_PrintPreview()
```

3 Call other print preview functions as needed.

See also

n_cst_dwsrv_report
u_dw

Functions

N_cst_dwsrv_printpreview contains precoded object functions:

of_FirstPage	of_Page
of_GetEnabled	of_PageAcross
of_GetInfo	of_PageCount
of_GetPropertyInfo	of_PageCountAcross
of_GetRuler	of_SetEnabled
of_GetZoom	of_SetRuler
of_LastPage	of_SetZoom

of_FirstPage

Description

Scrolls to the first page.

Access

Public

Syntax

dwcontrol.instancename.**of_FirstPage** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Not in n_cst_dssrv_printpreview
This function is not implemented in n_cst_dssrv_printpreview.

Examples

This example calls the of_FirstPage function:

```
dw_emplist.inv_printpreview.of_FirstPage( )
```

of_GetEnabled

Description

Reports whether the DataWindow is in print preview mode.

Access

Public

Syntax

dwcontrol.instancename.**of_GetEnabled** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Boolean. Returns TRUE if the DataWindow is in print preview mode and FALSE if it is not.

Examples

This example calls the of_GetEnabled function:

```
IF NOT dw_emp.inv_printpreview.of_GetEnabled() THEN
   dw_emp.inv_printpreview.of_SetEnabled(TRUE)
END IF
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_printpreview.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Not in n_cst_dssrv_printpreview
This function is not implemented in n_cst_dssrv_printpreview.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_printpreview.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
```

```
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRuler

Description

Reports whether a ruler is displayed.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRuler** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Boolean. Returns TRUE if the ruler is displayed and FALSE if it is not.

Examples

This example calls the of_GetRuler function:

```
IF NOT dw_emp.inv_printpreview.of_GetRuler ( ) THEN
   dw_emp.inv_printpreview.of_SetEnabled(TRUE)
END IF
```

of_GetZoom

Description

Retrieves the current zoom factor.

Access

Public

Syntax

dwcontrol.instancename.**of_GetZoom** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns the current zoom percentage if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetZoom function:

```
Integer   li_zoom

li_zoom = dw_emp.inv_printpreview.of_GetZoom( )
```

of_LastPage

Description

Scrolls to the last page.

Access

Public

Syntax

dwcontrol.instancename.**of_LastPage ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Not in n_cst_dssrv_printpreview
This function is not implemented in n_cst_dssrv_printpreview.

Examples

This example calls the of_LastPage function:

```
dw_emplist.inv_printpreview.of_LastPage( )
```

of_Page

Description

Retrieves the number of the current page.

Access

Public

Syntax

dwcontrol.instancename.**of_Page ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns the number of the current page if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Page function:

```
Integer   li_page

li_page = dw_emp.inv_printpreview.of_Page( )
```

of_PageAcross

Description

Retrieves the number of the current horizontal page.

Access

Public

Syntax

dwcontrol.instancename.**of_PageAcross ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns the number of the current horizontal page if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_PageAcross function:

```
Integer   li_page

li_page = dw_emp.inv_printpreview.of_PageAcross( )
```

of_PageCount

Description

Retrieves the total number of pages.

Access

Public

Syntax

dwcontrol.instancename.**of_PageCount ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns the total number of pages if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_PageCount function:

```
Integer   li_total

li_total = dw_emp.inv_printpreview.of_PageCount( )
```

of_PageCountAcross

Description

Retrieves the total number of horizontal pages.

Access

Public

Syntax

dwcontrol.instancename.**of_PageCountAcross** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)

Return value

Integer. Returns the total number of horizontal pages if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_PageCountAcross function:

```
Integer   li_total

li_total = &
   dw_emp.inv_printpreview.of_PageCountAcross( )
```

of_SetEnabled

Description

Controls a DataWindow's print preview setting.

Access

Public

Syntax

dwcontrol.instancename.**of_SetEnabled** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
boolean	Boolean specifying whether to place the DataWindow in print preview mode (TRUE) or normal view (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetEnabled function to place the dw_emp DataWindow into print preview mode:

```
String    ls_return

ls_return = &
   dw_emp.inv_printpreview.of_SetEnabled(TRUE)
IF ls_return <> "" THEN
   MessageBox("Print Preview Error", ls_return)
END IF
```

of_SetRuler

Description

Controls ruler display.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRuler** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
boolean	Boolean specifying whether to display the ruler (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetRuler function:

```
IF NOT dw_emp.inv_printpreview.of_GetRuler() THEN
   dw_emp.inv_printpreview.of_SetRuler(TRUE)
END IF
```

of_SetZoom

Description

Controls zoom factor.

Access

Public

Syntax

dwcontrol.instancename.**of_SetZoom** ({ *zoompercentage* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
zoompercentage (optional)	Integer specifying the DataWindow's zoom percentage. If you omit this argument, the w_zoom dialog box displays allowing the user to specify the zoom percentage

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetZoom function, displaying the w_zoom dialog box:

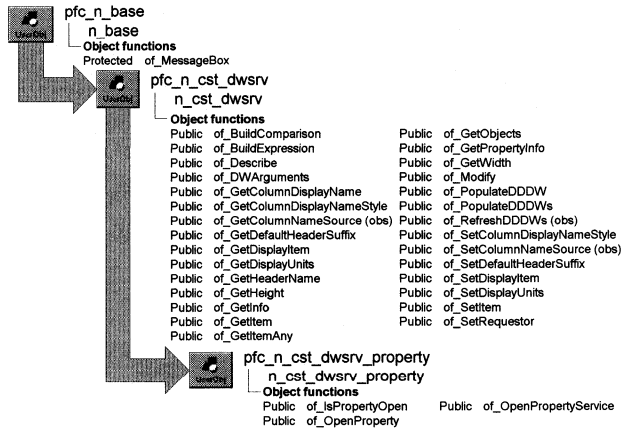
```
dw_emp.inv_printpreview.of_SetZoom( )
```

n_cst_dwsrv_property

Description

DataWindow property object.

Ancestry



Library

PFCUTIL.PBL
PFEUTIL.PBL

Object relationships

n_cst_dwpropertyattrib
n_cst_infoattrib
n_cst_metaclass
n_cst_propertyattrib
w_dwproperty
w_dwpropertyservice

Usage

U_dw uses this object to control the DataWindow Properties window.

See also

u_dw

Instance variables

Instance variable	Description	Data type	Access	Usage
iw_property	Reference to w_dwproperty window	w_dwproperty	Protected	Internal
iw_dwpropertyservices	Reference to w_dwpropertyservices window	w_dwpropertyservices	Protected	Internal

Events

N_cst_dwsrv_property includes one precoded event:

Destructor

Destructor

Description

Performs cleanup.

Usage

This event executes when the object is destroyed.

Functions

N_cst_debug includes precoded functions:

of_IsPropertyOpen	of_OpenPropertyService
of_OpenProperty	

of_IsPropertyOpen

Description

Reports whether either the w_dwproperty window or the w_dwpropertyservices dialog box is open.

Access

Public

Syntax

dwcontrol.instancename.**of_IsPropertyOpen** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)

Return value

Integer. Returns TRUE if either w_dwproperty or w_dwpropertyservices is open and FALSE if neither is open.

Examples

This example is from the u_dw RButtonUp event:

```
...
If IsValid(inv_property) Then
   If inv_property.of_IsPropertyOpen() = False Then
```

```
            lm_dw.m_table.m_debug.visible = True
            lm_dw.m_table.m_debug.enabled = True
        End If
...
```

of_OpenProperty

Description: Opens the w_dwproperty window.

Access: Public

Syntax: *dwcontrol.instancename*.**of_OpenProperty** (*properties*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)
properties	N_cst_dwpropertyattrib instance containing properties to display

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: Internal.

Examples: This example is from the u_dw pfc_Properties event:

```
...
If IsValid(inv_property) Then
      inv_property.of_OpenProperty(lnv_dwproperty)
   Else
      snv_property.of_OpenProperty(lnv_dwproperty)
   End If
...
```

of_OpenPropertyService

Description: Opens the w_dwpropertyservice dialog box for the specified service.

Access: Public

Syntax: *dwcontrol.instancename*.**of_OpenPropertyService** (*service*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)
service	PowerObject containing the service to be displayed in the w_dwpropertyservice dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_OpenPropertyService function:

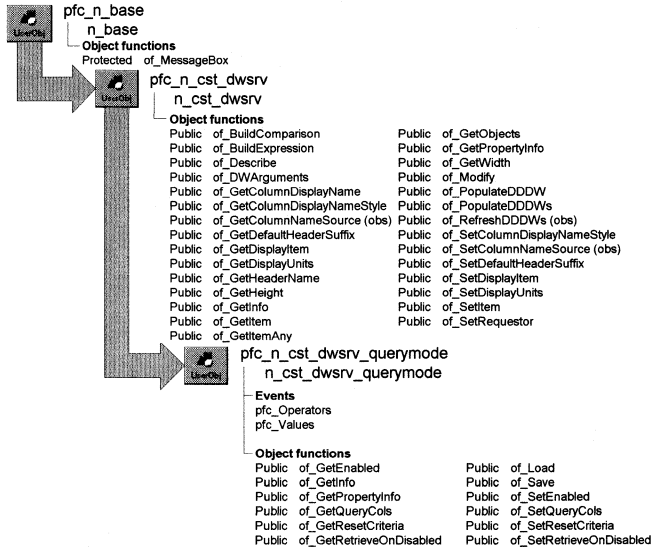
```
PowerObject  lpo_service

lpo_service = dw_1.inv_multitable
dw_1.inv_property.of_OpenPropertyService(lpo_service)
```

n_cst_dwsrv_querymode

Description

Service to enable and disable query mode. Includes the ability to save and load queries.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib
n_cst_selection

Usage

Use this object's functions to control DataWindow query mode processing. In particular, the of_Save and of_Load functions make it easy to access and save queries in a file.

To use this service:

1 Enable the service using the u_dw of_SetQuerymode function:

```
dw_emplist.of_SetQuerymode(TRUE)
```

2 Specify the columns eligible for query mode:

```
String    ls_cols[]

ls_cols[1] = "emp_dept_id"
ls_cols[2] = "emp_id"
```

```
dw_emplist.inv_querymode.of_SetQueryCols &
   (ls_cols)
```

3 Call query mode functions as needed.

Structures

Structure	Field	Data type
os_querymodeinfo	b_state	Boolean
	s_protect	String
	s_col	String

See also

n_cst_dwsrv_filter
n_cst_dwsrv_sort
u_dw

Instance variables

N_cst_dwsrv_querymode includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_resetquerycriteria	Indicates whether PFC resets query criteria between query mode toggles	Boolean	Protected	Internal
ib_retrieveondisabled	Indicates whether PFC automatically retrieves the DataWindow when query mode is disabled	Boolean	Protected	Internal
istr_querymode_info[ ]	Stores querymode information for each of the DataWindow's columns	os_querymodeinfo	Protected	Internal

Events

N_cst_dwsrv_querymode includes precoded events:

pfc_Operators
pfc_Values

pfc_Operators

Description — Displays a selection dialog box, prompting the user to select an operator (=, >=, <=, and so on) for use in query mode criteria.

Return value — Integer. Returns 1 if the event succeeds, 0 if the user cancels out of the selection dialog box, and -1 if an error occurs.

Usage — This event is called when the user clicks the right mouse button over the DataWindow and selects Operators from the m_dw popup menu.

pfc_Values

Description — Displays a selection dialog box, prompting the user to select a value for use in query mode criteria. The dialog box displays a list of all unique values for the clicked column.

Return value — Integer. Returns 1 if the event succeeds, 0 if the user cancels out of the selection dialog box, and -1 if an error occurs.

Usage — This event is called when the user clicks the right mouse button over a DataWindow column and selects Values from the m_dw popup menu.

Functions

N_cst_dwsrv_querymode contains precoded object functions:

of_GetEnabled	of_Load
of_GetInfo	of_Save
of_GetPropertyInfo	of_SetEnabled
of_GetQueryCols	of_SetQueryCols
of_GetResetCriteria	of_SetResetCriteria
of_GetRetrieveOnDisabled	of_SetRetrieveOnDisabled

of_GetEnabled

Description — Retrieves the DataWindow's current query mode state.

Access — Public

Syntax — *dwcontrol.instancename*.**of_GetEnabled** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)

Return value

Boolean. Returns TRUE if query mode is currently enabled and FALSE if it is not.

Usage

Enable and disable query mode by calling the of_SetEnabled function.

Examples

This example calls the of_GetEnabled function:

```
IF NOT dw_emp.inv_querymode.of_GetEnabled() THEN
   dw_emp.inv_querymode.of_SetEnabled(TRUE)
END IF
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_querymode.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
```

```
        + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description: Retrieves information about the service's properties.

Access: Public

Syntax: *dwcontrol.instancename.***of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: The DataWindow Properties window calls this function to access service information:

Examples: This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_querymode.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetQueryCols

Description: Retrieves the columns that are eligible for query criteria.

Access: Public

Syntax: *dwcontrol.instancename.***of_GetQueryCols** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
columns	String array containing the columns that can be set as querymode columns (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetQueryCols function:

```
String  ls_cols[ ]
Integer  li_count, li_loop

li_count =  &
  dw_1.inv_querymode.of_GetQueryCols  &
    (ls_cols)
lb_1.Reset()
FOR li_loop = 1 to li_count
   lb_1.AddItem(ls_cols[li_loop])
NEXT
```

of_GetResetCriteria

Description

Reports whether PFC resets query criteria between query mode toggles.

Access

Public

Syntax

dwcontrol.instancename.**of_GetResetCriteria** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)

Return value

Boolean. Returns TRUE if PFC resets query criteria between query mode toggles and FALSE if it does not.

Usage

Call this function to determine whether PFC resets query criteria between query mode toggles.

Examples

This example calls the of_GetResetCriteria function:

```
Boolean   lb_reset
```

```
lb_reset = &
   dw_emp.inv_querymode.of_GetResetCriteria( )
```

of_GetRetrieveOnDisabled

Description

Reports whether PFC automatically retrieves the DataWindow when query mode is disabled.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRetrieveOnDisabled** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)

Return value

Boolean. Returns TRUE if PFC automatically retrieves the DataWindow when query mode is disabled and FALSE if it does not.

Usage

Call this function to determine whether PFC automatically retrieves the DataWindow when query mode is disabled.

Examples

This example calls the of_GetRetrieveOnDisabled function:

```
Boolean   lb_retrieve

lb_retrieve = &
   dw_emp.inv_querymode.of_GetRetrieveOnDisabled( )
```

of_Load

Description

Displays the File Open dialog box, prompting the user to select a saved query from disk. If the user selects a file, this function enables query mode for the DataWindow and uses the selected file to determine selection criteria.

Access

Public

Syntax

dwcontrol.instancename.**of_Load** (*title*, *pathname*, *filename* {, *extension* {, *filter* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
title	String specifying the title of the File Open dialog box
pathname	String into which the function returns the path for the chosen file (passed by reference)
filename	String into which the function returns the filename (passed by reference)
extension (optional)	String specifying the 1- to 3-character default file extension. The default is TXT
filter (optional)	String specifying a text description of the files to include in the listbox and the file mask that you want to use to select the displayed files (for example, *.* or *.exe). The format for filter is: description,*.ext. The default is: "Queries (*.TXT),*.TXT"

Return value

Integer. Returns 1 if the function succeeds, 0 if the user clicks Cancel or the system cancels the display, and -1 if an error occurs.

Usage

Call this function to allow the user to select a previously saved query.

Call the of_Save function to save a query.

Examples

This example calls the of_Load function:

```
Integer   li_return
String    ls_path_name, ls_doc_name

ls_path_name = "c:\temp\rkn.txt"
dw_employee.inv_querymode.of_Load &
   ("Load a Query", ls_path_name, ls_doc_name)
```

of_Save

Description

Displays the File Save dialog box, allowing the user to save a query to disk. This function works only when the DataWindow is in query mode.

Access

Public

Syntax

dwcontrol.instancename.**of_Save** (*title*, *pathname*, *filename* {, *extension* {, *filter* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
title	String specifying the title of the File Save dialog box
pathname	String specifying the default path and filename. The function uses this argument to return the path for the saved file (passed by reference)
filename	String into which the function returns the filename (passed by reference)
extension (optional)	String specifying the 1- to 3-character default file extension. The default is TXT
filter (optional)	String specifying a text description of the files to include in the listbox and the file mask that you want to use to select the displayed files (for example, *.* or *.exe). The format for filter is: description,*.ext. The default is: "Queries (*.TXT),*.TXT"

Return value

Integer. Returns 1 if the function succeeds, 0 if the user clicks Cancel or the system cancels the display, and -1 if an error occurs.

Usage

Call this function to allow the user to select a previously saved query.

Call the of_Load function to use a previously saved query.

The query must have been applied before it can be saved.

Examples

This example calls the of_Save function:

```
String    ls_path_name, ls_doc_name

ls_path_name = "c:\temp\rkn.txt"
dw_employee.inv_querymode.of_Save &
   ("Save a Query", ls_path_name, ls_doc_name)
```

of_SetEnabled

Description

Controls a DataWindow's query mode setting. This function also removes the border and protects non-query mode columns.

Access

Public

Syntax

dwcontrol.instancename.**of_SetEnabled** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
boolean	Boolean specifying whether to turn query mode on (TRUE) or off (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetEnabled function:

```
Integer   li_return

li_return = dw_master.inv_querymode.of_SetEnabled &
   (NOT dw_master.inv_querymode.of_GetEnabled( ) )
```

of_SetQueryCols

Description

Specifies the columns that are eligible for query criteria and changes the tab order and border style for those columns

Access

Public

Syntax

dwcontrol.instancename.**of_SetQueryCols** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
columns	String array specifying the columns to be set as querymode columns

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This calls the of_SetQueryCols function:

```
Integer   li_itemtotal, li_itemcount
Integer   li_sel_cnt
String    ls_cols[ ]

// Get the number of items in the ListBox.
```

```
li_itemtotal = lb_col.TotalItems( )

// Loop through all the items.
FOR li_itemcount = 1 TO li_itemtotal
   IF lb_col.State(li_itemcount) = 1 THEN
      li_sel_cnt++
      ls_cols[li_sel_cnt] = &
      lb_col.Text(li_itemcount)
   END IF
NEXT

dw_emp.inv_querymode.of_SetQueryCols &
   (ls_cols)
```

of_SetResetCriteria

Description

Specifies whether PFC resets query criteria between query mode toggles.

Access

Public

Syntax

dwcontrol.instancename.**of_SetResetCriteria** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
boolean	Boolean indicating whether PFC resets query criteria (TRUE) or not (FALSE)

Return value

Boolean. Returns TRUE if PFC resets query criteria between query mode toggles and FALSE if it does not.

Examples

This example calls the of_SetResetCriteria function:

```
dw_emp.inv_querymode.of_SetResetCriteria(TRUE)
```

of_SetRetrieveOnDisabled

Description

Specifies whether PFC automatically retrieves the DataWindow when query mode is disabled.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRetrieveOnDisabled** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_qucrymode)
boolean	Boolean indicating whether PFC automatically retrieves the DataWindow (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to specify whether PFC automatically retrieves the DataWindow when query mode is disabled.

Examples

This example calls the of_SetRetrieveOnDisabled function:

```
dw_emp.inv_querymode.of_SetRetrieveOnDisabled(TRUE)
```

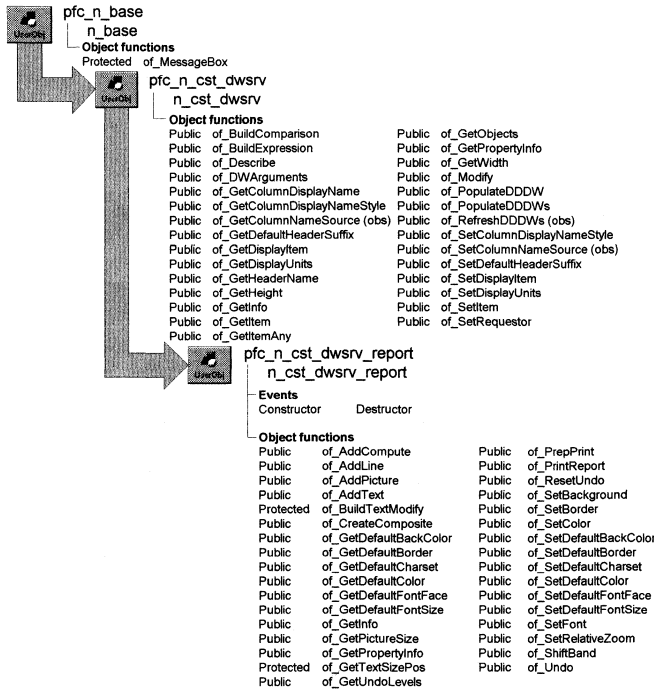
n_cst_dwsrv_report

Description

Reporting service. This object contains functions that help you handle report DataWindows and their printing.

N_cst_dssrv_report is the same as n_cst_dwsrv_report
N_cst_dssrv_report, the DataStore report service object, is almost identical to n_cst_dwsrv_report except that it applies to DataStores instead of DataWindow controls. Except where noted, references to n_cst_dwsrv_report in the following discussions apply equally to n_cst_dssrv_report.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_platform
n_cst_propertyattrib
n_cst_string

Usage

Use this service with report DataWindows to provide enhanced viewing and printing capabilities.

Many of this service's functions provide the option of either executing the DataWindow Modify function or returning Modify syntax for use as input to your own Modify function. If you code more than two consecutive report service functions, consider returning the Modify syntax, concatenating the strings, and issuing the Modify function from within your own code.

To use this service, the DataWindow object must use PBUs or pixels as the DataWindow unit. It does not work with DataWindows that use Thousandths of an Inch or Thousandths of a Centimeter as the DataWindow unit.

To use this service:

1 Enable the service using the u_dw of_SetReport function:

```
dw_emp.of_SetReport(TRUE)
```

2 Call reporting and printing functions as needed.

See also

n_cst_dwsrv_printpreview
u_dw

Instance variables

N_cst_dwsrv_report contains instance variables:

Instance variable	Description	Data type	Access	Usage
ibo_defaultborder	Default border	Border	Protected	Internal
ifc_defaultcharset	Default font character set	FontCharSet	Protected	Internal
ii_defaultfontsize	Default font size	Integer	Protected	Internal
ii_objectnum	Internal counter of the objects created in the associated DataWindow	Integer	Protected	Internal (updated by of_Add functions)
ii_undolevel	Counter of the number of changes made to the DataWindow	Integer	Protected	Internal
il_defaultbackcolor	Default background color	Long	Protected	Internal
il_defaultcolor	Default color	Long	Protected	Internal

Instance variable	Description	Data type	Access	Usage
inv_platform	Reference variable for platform services	n_cst_platform	Protected	Internal
is_undo[ ]	Array containing changed or deleted DataWindow columns and text fields	String	Protected	Internal (used by of_ResetUndo and of_Undo)

Events

N_cst_dwsrv_report includes precoded events:

Constructor
Destructor

Constructor

Description — Sets defaults for the report service.

Usage — This event executes when the DataWindow is created.

Destructor

Description — Destroys the platform service instance (inv_platform instance variable).

Usage — This event executes when the DataWindow is destroyed.

Functions

N_cst_dwsrv_report contains precoded functions:

of_AddCompute	of_PrepPrint
of_AddLine	of_PrintReport
of_AddPicture	of_ResetUndo
of_AddText	of_SetBackground
of_BuildTextModify	of_SetBorder
of_CreateComposite	of_SetColor
of_GetDefaultBackColor	of_SetDefaultBackColor

of_GetDefaultBorder	of_SetDefaultBorder
of_GetDefaultCharset	of_SetDefaultCharset
of_GetDefaultColor	of_SetDefaultColor
of_GetDefaultFontFace	of_SetDefaultFontFace
of_GetDefaultFontSize	of_SetDefaultFontSize
of_GetInfo	of_SetFont
of_GetPictureSize	of_SetRelativeZoom
of_GetPropertyInfo	of_ShiftBand
of_GetTextSizePos	of_Undo
of_GetUndoLevels	

of_AddCompute

Description

Adds a computed column to any band of a DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_AddCompute** (*expression, band, xcoordinate, ycoordinate* {, *border* {, *bold, italic, underline* {, *fontface, fontsize, characterset* {, *color, backcolor* {, *execute* } } } } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
expression	String specifying the expression to be added as the computed column
band	String specifying the band to which the function will add the computed column. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
xcoordinate	Either of the following: ♦ Integer specifying the *x* location of the computed column. If you specify an integer for *xcoordinate*, you must also specify an integer for *ycoordinate* ♦ Alignment enumerated data type (Center!, Justify!, Left!, or Right!). If you specify an alignment enumerated data type for *xcoordinate*, you must specify a VTextAlign enumerated data type for *ycoordinate*

Argument	Description
ycoordinate	Either of the following: ♦ Integer specifying the *y* location of the computed column. If you specify an integer for *ycoordinate*, you must also specify an integer for *xcoordinate* ♦ VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!). If you specify a VTextAlign enumerated data type for *ycoordinate*, you must specify an Alignment enumerated data type for *xcoordinate*
border (optional)	Border enumerated data type specifying the border style: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised! Specify the default border style using the of_SetDefaultBorder function
bold (optional)	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default)
italic (optional)	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default)
underline (optional)	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default)
fontface (optional)	String specifying the font to use Specify the default font face using the of_SetDefaultFontFace function
fontsize (optional)	Integer specifying the point size of the font for the computed column Specify the default font size using the of_SetDefaultFontSize function

Argument	Description
characterset (optional)	FontCharSet enumerated data type specifying the character set to use for the computed column: ANSI! Symbol! ChineseBig5! Hangul! ShiftJIS! OEM! Specify the default character set using the of_SetDefaultCharset function
color (optional)	Long specifying the color of the computed field Specify the default color using the of_SetDefaultColor function
backcolor (optional)	Long specifying the background color of the computed field Specify the default background color using the of_SetDefaultBackColor function
execute (optional)	Boolean specifying whether to execute the Modify function that adds the computed column to the band (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the computed column; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the computed column.

Usage

Depending on the *execute* argument, the function either adds the computed column to the DataWindow or returns syntax that you use in a Modify function to add the computed column.

On Macintosh
This function is not available on the Macintosh.

Examples

This example adds a computed column to display the page number in the bottom-right corner of the footer:

```
String    ls_return, ls_expression

ls_expression = "~"Page ~" + page()"
ls_Return = dw_employee.inv_report.of_AddCompute &
   (ls_expression, "footer", Right!, Bottom!, &
```

```
        Underline!, TRUE, FALSE, FALSE, "Arial", &
           14, ANSI!)

IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
END IF
```

of_AddLine

Adds a line to any band of a DataWindow. There are two syntaxes.

To	Use
Add a line by specifying the x and y coordinates of the endpoints	Syntax 1
Add a line by specifying a VTextAlign enumerated data type	Syntax 2

Syntax 1

Add a line by specifying the x and y coordinates of the endpoints

Description

Adds a line by specifying the x and y coordinates of the endpoints.

Access

Public

Syntax

dwcontrol.instancename.**of_AddLine** (*band, penstyle, penwidth, x1, y1, x2, y2* {, *color, backcolor* {, *execute* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
band	String specifying the band to which the function will add the line. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*

Argument	Description
penstyle	Linestyle enumerated data type specifying the pen style of the line: Continuous! Dash! Dot! DashDot! DashDotDot! Transparent!
penwidth	Integer specifying the pen width of the line
x1	Integer specifying the *x1* location of the line
y1	Integer specifying the *y1* coordinate of the line
x2	Integer specifying the *x2* location of the line
y2	Integer specifying the *y2* coordinate of the line
color (optional)	Long specifying the color of the line Specify the default color using the of_SetDefaultColor function
backcolor (optional)	Long specifying the background color of the line Specify the default background color using the of_SetDefaultBackColor function
execute (optional)	Boolean specifying whether to execute the Modify function that adds the line to the band (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the line; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the line.

Usage

Depending on the *execute* argument, the function either adds the line to the DataWindow or returns syntax that you use in a Modify function to add the line.

You can call this function to draw a separator line above the footer area when preparing a DataWindow for printing.

Use Syntax 2 to add a line by specifying a VTextAlign enumerated data type.

Examples

This example adds a line to the top of the DataWindow's footer:

```
String   ls_return, ls_expression
```

```
// Draw the line.
ls_Return = dw_employee.inv_report.of_AddLine &
   ("footer", Continuous!, 10, 1, 1, 1, 100)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
   Return
END IF
```

Syntax 2 **Add a line by specifying a VTextAlign enumerated data type**

Description

Adds a line by specifying the a VTextAlign enumerated data type.

Access

Public

Syntax

dwcontrol.instancename.**of_AddLine** (*band, penstyle, penwidth, placement* {, *color, backcolor* {, *execute* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
band	String specifying the band to which the function will add the line. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
penstyle	Linestyle enumerated data type specifying the pen style of the line: Continuous! Dash! Dot! DashDot! DashDotDot! Transparent!
penwidth	Integer specifying the pen width of the line

Argument	Description
placement	VTextAlign enumerated data type specifying the line placement: Bottom! MultiLine! Top! VCenter!
color (optional)	Long specifying the color of the line Specify the default color using the of_SetDefaultColor function
backcolor (optional)	Long specifying the background color of the line Specify the default background color using the of_SetDefaultBackColor function
execute (optional)	Boolean specifying whether to execute the Modify function that adds the line to the band (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the line; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the line.

Usage

Depending on the *execute* argument, the function either adds the line to the DataWindow or returns syntax that you use in a Modify function to add the line.

You can call this function to draw a separator line above the footer area when preparing a DataWindow for printing.

Use Syntax 1 to add a line by specifying *x* and *y* coordinates.

Examples

This example adds a line to the top of the DataWindow's footer:

```
String   ls_return, ls_expression

// Draw the line.
ls_Return = dw_employee.inv_report.of_AddLine &
   ("footer", Continuous!, 10, Top!)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
   Return
END IF
```

of_AddPicture

Description

Adds a picture object to the specified DataWindow band.

Access

Public

Syntax

dwcontrol.instancename.**of_AddPicture** (*filename, band, xcoordinate, ycoordinate* {, *border* {, *execute* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
filename	String specifying the name of the bitmap file If the directory containing the file is not in your system path, you must specify the complete path
band	String specifying the name of the band to which the function will add the bitmap. Valid bands are: ♦ Header ♦ Detail ♦ Footer ♦ Summary ♦ Header.*levelnumber* ♦ Trailer.*levelnumber*
xcoordinate	Either of the following: ♦ Integer specifying the *x* location of the bitmap. If you specify an integer for *xcoordinate*, you must also specify an integer for *ycoordinate* ♦ Alignment enumerated data type (Center!, Justify!, Left!, or Right!). If you specify an alignment enumerated data type for *xcoordinate*, you must specify a VTextAlign enumerated data type for *ycoordinate*
ycoordinate	Either of the following: ♦ Integer specifying the *y* location of the bitmap. If you specify an integer for *ycoordinate*, you must also specify an integer for *xcoordinate* ♦ VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!). If you specify a VTextAlign enumerated data type for *ycoordinate*, you must specify an Alignment enumerated data type for *xcoordinate*

Argument	Description
border (optional)	Border enumerated data type specifying the border style: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised! Specify the default border style using the of_SetDefaultBorder function
execute (optional)	Boolean specifying whether to execute the Modify function that adds the bitmap to the band (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the bitmap; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the bitmap.

Usage

Depending on the *execute* argument, the function either adds the bitmap to the DataWindow or returns syntax that you use in a Modify function to add the bitmap.

Examples

This example adds the logo.bmp bitmap to the Header band:

```
String    ls_return

ls_Return = dw_employee.inv_report.of_AddPicture &
   ("logo.bmp", "header", Center!, Top!, &
      NoBorder!, TRUE)

IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
END IF
```

of_AddText

Description

Adds text to any band of a DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_AddText** (*text, band, xcoordinate, ycoordinate* {, *border* {, *bold, italic, underline* {, *fontface, fontsize, characterset* {, *color, backcolor* {,*execute* } } } } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
text	String specifying the text to be added
band	String specifying the band to which the function will add the text. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
xcoordinate	Either of the following: ♦ Integer specifying the *x* location of the text. If you specify an integer for *xcoordinate*, you must also specify an integer for *ycoordinate* ♦ Alignment enumerated data type (Center!, Justify!, Left!, or Right!). If you specify an alignment enumerated data type for *xcoordinate*, you must specify a VTextAlign enumerated data type for *ycoordinate*
ycoordinate	Either of the following: ♦ Integer specifying the *y* location of the text. If you specify an integer for *ycoordinate*, you must also specify an integer for *xcoordinate* ♦ VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!). If you specify a VTextAlign enumerated data type for *ycoordinate*, you must specify an Alignment enumerated data type for *xcoordinate*

Argument	Description
border (optional)	Border enumerated data type specifying the border style: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised! Specify the default border style using the of_SetDefaultBorder function
bold (optional)	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default)
italic (optional)	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default)
underline (optional)	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default)
fontface (optional)	String specifying the font to use Specify the default font face using the of_SetDefaultFontFace function
fontsize (optional)	Integer specifying the point size of the font for the text Specify the default font size using the of_SetDefaultFontSize function
characterset (optional)	FontCharSet enumerated data type specifying the character set to use for the computed column: ANSI! Symbol! ChineseBig5! Hangul! ShiftJIS! OEM! Specify the default character set using the of_SetDefaultCharset function
color (optional)	Long specifying the color of the text Specify the default color using the of_SetDefaultColor function

Argument	Description
backcolor (optional)	Long specifying the background color of the text Specify the default background color using the of_SetDefaultBackColor function
execute (optional)	Boolean specifying whether to execute the Modify function that adds the text to the band (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the text; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the text.

Usage

Depending on the *execute* argument, the function either adds the text to the DataWindow or returns syntax that you use in a Modify function to add the text.

On Macintosh
This function is not available on the Macintosh.

Examples

This example adds text to the DataWindow's header area:

```
String    ls_return

ls_return = dw_employee.inv_report.of_AddText &
   ("**Company Confidential**", "header", Center!, &
      Top!, Raised!, TRUE, FALSE, FALSE)
IF Trim(ls_return) <> "" THEN
   MessageBox("Modify Error", ls_return)
END IF
```

of_BuildTextModify

Description

Builds input text for the PowerScript Modify function.

Access

Protected

Syntax

dwcontrol.instancename.**of_BuildTextModify** (*type, text, band, xcoordinate, ycoordinate, height, width, border, bold, italic, underline, fontface, fontsize, characterset, color, backcolor*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
type	String specifying the type of object being added (text or computed column)
text	String specifying the text to be added
band	String specifying the band to which the function will add the text. Valid bands are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
xcoordinate	Integer specifying the *x* location of the text
ycoordinate	Integer specifying the *y* location of the text
height	Integer specifying the height of the text
width	Integer specifying the width of the text
border	Border enumerated data type specifying the border style: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised! Specify the default border style using the of_SetDefaultBorder function
bold	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default)
italic	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default)

Argument	Description
underline	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default)
fontface	String specifying the font to use Specify the default font face using the of_SetDefaultFontFace function
fontsize	Integer specifying the point size of the font Specify the default font size using the of_SetDefaultFontSize function
characterset	FontCharSet enumerated data type specifying the character set to use: ANSI! Symbol! ChineseBig5! Hangul! ShiftJIS! OEM! Specify the default character set using the of_SetDefaultCharset function
color	Long specifying the color Specify the default color using the of_SetDefaultColor function
backcolor	Long specifying the background color Specify the default background color using the of_SetDefaultBackColor function

Return value

String. Returns the text of the Modify command to add the object.

Usage

Internal.

Examples

This example is from the of_AddCompute function:

```
...
ls_modify = ls_modify + of_BuildTextModify &
   ("compute", as_expr, as_band, ai_x, ai_y, &
      li_height, li_width, abo_border, ab_bold, &
         ab_italic, ab_underline, as_fontface, &
            ai_fontsize, afc_charset, al_color, &
               al_backcolor)
...
```

of_CreateComposite

Description

Creates a composite DataWindow from an array of DataWindows. Composite DataWindows enable you to print more than one DataWindow on a page.

Access

Public

Syntax

dwcontrol.instancename.**of_CreateComposite** (*datawindows, arrangevertical, trailfooter, slide, border*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
datawindows	String array naming the DataWindow objects to make up the composite DataWindow
arrangevertical	Boolean specifying whether to arrange the DataWindows vertically (TRUE) or horizontally (FALSE)
trailfooter	String array of values to set the trail_footer property for each DataWindow. Valid values are: Yes No An empty string (defaults to Yes)
slide	String array of values to set the Slide properties for each DataWindow. If *arrangevertical* is TRUE, this function sets the DataWindow's SlideUp property. Valid values are: AllAbove DirectlyAbove No An empty string (defaults to AllAbove) If *arrangevertical* is FALSE, this function sets the SlideLeft property. Valid values are: Yes No

Argument	Description
border	Array of the Border enumerated data type specifying the border for each DataWindow in the composite: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised!

Return value

Integer. Returns 1 if the function succeeds and -1 if it fails.

Examples

This example creates a composite report containing the d_employee and d_benefits DataWindows:

```
String    ls_dws[ ], ls_trailfooter[ ], ls_slide[ ]
String    ls_return
Integer   li_return
Boolean   lb_vertical
Border    lbo_border[ ]

lb_vertical = TRUE
ls_dws[1] = "d_employee"
ls_dws[2] = "d_benefits"
ls_trailfooter[1] = "No"
ls_trailfooter[2] = "Yes"
ls_slide[1] = "AllAbove"
ls_slide[2] = "AllAbove"
lbo_border[1] = Lowered!
lbo_border[2] = Lowered!

li_Return = &
   dw_composite.inv_report.of_CreateComposite &
      (ls_dws, lb_vertical, ls_trailfooter, &
         ls_slide, lbo_border)
IF li_Return = 1 THEN
   dw_composite.SetTransObject(SQLCA)
   dw_composite.Event pfc_Retrieve( )
END IF
```

of_GetDefaultBackColor

Description

Retrieves the default background color.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDefaultBackColor** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

Long. Returns the default background color.

Examples

This example calls the of_GetDefaultBackColor function:

```
Long   ll_backcolor

ll_backcolor = &
   dw_emp.inv_report.of_GetDefaultBackColor( )
```

of_GetDefaultBorder

Description

Retrieves the default border.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDefaultBorder** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

Border. Returns a Border enumerated data type identifying the border:

NoBorder!
ShadowBox!
Box!
ResizeBorder!
Underline!
Lowered!
Raised!

Examples

This example calls the of_GetDefaultBorder function:

```
Border   lbo_border

lbo_border = &
   dw_emp.inv_report.of_GetDefaultBorder( )
IF lb_border = Box! THEN
...
```

of_GetDefaultCharset

Description

Retrieves the default character set.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDefaultCharset ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

FonCharSet. Returns a FontCharSet enumerated data type identifying the default character set:

ANSI!
Symbol!
ChineseBig5!
Hangul!
ShiftJIS!
OEM!

Examples

This example calls the of_GetDefaultCharset function:

```
FontCharSet   lfc_charset
lfc_charset = &
   dw_emp.inv_report.of_GetDefaultCharset()
IF lfc_charset = ANSI! THEN
...
```

of_GetDefaultColor

Description

Retrieves the default color.

Access

Public

Syntax

dwcontrol.instancename.**of_GetDefaultColor ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

Long. Returns the default color.

Examples

This example calls the of_GetDefaultColor function:

```
Long   ll_color

ll_color = &
   dw_emp.inv_report.of_GetDefaultColor( )
```

of_GetDefaultFontFace

Description

Retrieves the default font (for example, Arial).

Access

Public

Syntax

dwcontrol.instancename.**of_GetDefaultFontFace ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

String. Returns the default font.

Examples

This example calls the of_GetDefaultFontFace function:

```
String    ls_font

ls_font = dw_emp.inv_report.of_GetDefaultFontFace()
IF ls_font = "Arial" THEN
...
```

of_GetDefaultFontSize

Description: Retrieves the default font point size.

Access: Public

Syntax: *dwcontrol.instancename*.**of_GetDefaultFontSize** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value: Integer. Returns the default font size.

Examples: This example calls the of_GetDefaultFontSize function:

```
Integer   li_fontsize
li_fontsize = &
   dw_emp.inv_report.of_GetDefaultFontSize()
...
```

of_GetInfo

Description: Retrieves service information.

Access: Public

Syntax: *dwcontrol.instancename*.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default for this is inv_report)
infoobject	n_cst_infoattrib instance into which the function places service information (passed by reference)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: The DataWindow Properties window calls this function to access service information.

Examples: This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_report.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPictureSize

Description

Calculates the height and width of a picture object in PBUs or pixels. A PowerBuilder picture object can display a bitmap (BMP) file, a runlength-encoded (RLE) file, or a Windows metafile (WMF).

Access

Public

Syntax

dwcontrol.instancename.**of_GetPictureSize** (*filename, height, width*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
filename	String specifying the name of the file containing the picture. This must be either a fully qualified filename or reside in a directory on the system path
height	Integer specifying the height of the picture in PBUs or pixels (passed by reference)
width	Integer specifying the width of the picture in PBUs or pixels (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example returns the height and width for the logo.bmp file. The function returns the height in li_height and the width in li_width:

```
Integer   li_return,li_height,li_width

li_return = &
   dw_employee.inv_report.of_GetPictureSize &
      ("logo.bmp", li_height, li_width)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default for this is inv_report)
propertyobject	n_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Not in n_cst_dssrv_report
This function is not implemented in n_cst_dssrv_report.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_report.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetTextSizePos

Description

Calculates the size and position of a text object.

Access

Protected

Syntax

dwcontrol.instancename.**of_GetTextSizePos** (*text, band, halign, valign, bold, italic, underline, fontface, fontsize, height, width, xcoordinate, ycoordinate*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
text	String specifying the text to be added
band	String specifying the band. Valid values are: Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
halign	Alignment enumerated data type (Center!, Justify!, Left!, or Right!) specifying horizontal alignment
valign	VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!) specifying vertical alignment
bold	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default)
italic	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default)
underline	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default)
fontface	String specifying the font to use. Specify the default font face using the of_SetDefaultFontFace function
fontsize	Integer specifying the point size of the font. Specify the default font size using the of_SetDefaultFontSize function
height	Integer specifying the height of the text
width	Integer specifying the width of the text
xcoordinate	Integer specifying the *x* location of the text
ycoordinate	Integer specifying the *y* location of the text

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

On Macintosh
This function is not available on the Macintosh.

Examples

This example is from the of_AddCompute function:

```
...
li_newlines = of_GetTextSizePos(as_expr, as_band, &
   aal_halign, avta_valign, ab_bold, ab_italic, &
      ab_underline, as_fontface, ai_fontsize, &
         li_height, li_width, ai_x, ai_y)
...
```

of_GetUndoLevels

Description

Retrieves the DataWindow's current undo level.

Access

Public

Syntax

dwcontrol.instancename.**of_GetUndoLevels ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

Integer. Returns the number of changes to the DataWindow.

Usage

Call this function when determining the number of levels to specify when using the of_Undo function.

Examples

This example calls the of_GetUndoLevels function to display the number of undo levels in the status bar:

```
gnv_app.of_GetFrame( ).SetMicroHelp &
   ("Undo Levels: " + &
      String(dw_emp.inv_report.of_GetUndoLevels( )))
```

of_PrepPrint

Description

Prepares a DataWindow for printing: changes all colors to black on white and removes borders from all columns.

Access

Public

Syntax

dwcontrol.instancename.**of_PrepPrint** (*underlineheadings*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
underlineheadings	Boolean specifying whether to replace header borders with underlines (TRUE) or remove header borders (FALSE) This function only changes column headings whose object names end with _t

Return value

String. Returns the output of the Modify command. An empty string ("") indicates success.

Usage

Call this function to modify a DataWindow to a format more suitable for printing (black on white; no borders around columns).

This function removes borders from all columns, including headers. You can highlight headings by setting *underlineheadings* to TRUE, which replaces header borders with underlines.

This function is called by of_PrintReport and calls of_SetColor and of_SetBorder.

Examples

This example calls of_PrepPrint before printing dw_employee using underlined headings:

```
String    ls_return

// Prepare for printing.
ls_Return = &
   dw_employee.inv_report.of_PrepPrint(TRUE)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Print Error", &
      "Error preparing DataWindow for printing")
END IF
// Print report.
dw_employee.Print(TRUE)
```

of_PrintReport

Description

Prepares a DataWindow for printing, prints it, and restores it to its original state. This function calls of_PrepPrint.

Access

Public

Syntax

dwcontrol.instancename.**of_PrintReport** (*underlineheadings* {, *canceldialog* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
underlineheadings	Boolean specifying whether to replace header borders with underlines (TRUE) or remove header borders (FALSE) This function only changes column headings whose object names end with _t
canceldialog (optional)	Boolean specifying whether to display the Cancel dialog box (TRUE) or not (FALSE). The default is FALSE

Return value

String. Returns the output of the Modify command. An empty string ("") indicates success.

Examples

This example prints dw_employee with underlined headings:

```
String   ls_return

ls_return = &
   dw_employee.inv_report.of_PrintReport(TRUE,TRUE)
IF ls_Return <> "" THEN
   MessageBox("Print Error", &
      "Error printing DataWindow")
END IF
```

of_ResetUndo

Description

Resets the number of undo levels to zero.

Access

Public

Syntax

dwcontrol.instancename.**of_ResetUndo** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Return value

None

Usage

Call this function to clear the is_undo array, which maintains a list of changes to the DataWindow.

Examples

This example uses the of_ResetUndo function to reset the number of undo levels for dw_employee:

```
dw_employee.inv_report.of_ResetUndo()
```

of_SetBackground

Places a picture in the background of the DataWindow. There are two syntaxes:

To	Use
Add a background picture at its exact size	Syntax 1
Add a background picture, sizing it to fit the DataWindow	Syntax 2

Syntax 1

Add a background picture at its exact size

Description

Add a background picture, displayed at its exact size.

Access

Public

Syntax

dwcontrol.instancename.**of_SetBackground** (*filename* {, *xcoordinate* {, *ycoordinate* {, *height*, *width* {, *execute* } } } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
filename	String specifying the name of the bitmap file If the directory containing the file is not in your system path, you must specify the complete path
xcoordinate (optional)	Integer specifying the *x* location of the bitmap. The default is 0

Argument	Description
ycoordinate (optional)	Integer specifying the *y* location of the bitmap. The default is 0
height (optional)	Integer specifying the height of the bitmap being placed. If you pass 0, the function uses the of_GetPictureSize function to calculate this value. The default is 0
width (optional)	Integer specifying the width of the bitmap being placed. If you pass 0, the function uses the of_GetPictureSize function to calculate this value. The default is 0
execute (optional)	Boolean specifying whether to execute the Modify function that adds the background picture (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the background picture; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the background picture.

Usage

A background picture to can provide a watermark effect, enhancing a DataWindow's online and printed appearance.

Depending on the *execute* argument, the function either adds the background to the DataWindow or returns syntax that you use in a Modify function to add the background picture.

Use Syntax 2 to add a background picture, sizing it to fit the DataWindow.

Examples

This example uses the of_SetBackground function to add the confid.bmp background picture to dw_employee:

```
String    ls_return

ls_Return = &
   dw_employee.inv_report.of_SetBackground &
      ("confid.bmp", 20, 20)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify error",ls_Return)
END IF
```

Syntax 2

Add a background picture, sizing it to fit the DataWindow

Description

Add a background picture, optionally sizing it to fit the DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_SetBackground** (*filename* , *sizetofit*, *xcoordinate*, *ycoordinate*, *height*, *width execute*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
filename	String specifying the name of the bitmap file If the directory containing the file is not in your system path, you must specify the complete path
sizetofit	Boolean specifying whether to set the picture width to the width of the DataWindow and use the *height*/*width* ratio to set the height (TRUE) or use the *height* and *width* values
xcoordinate	Integer specifying the *x* location of the bitmap
ycoordinate	Integer specifying the *y* location of the bitmap
height	Integer specifying the height of the bitmap being placed. If you pass 0, the function uses the of_GetPictureSize function to calculate this value
width	Integer specifying the width of the bitmap being placed. If you pass 0, the function uses the of_GetPictureSize function to calculate this value
execute	Boolean specifying whether to execute the Modify function that adds the background picture (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that added the background picture; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to add the background picture.

Usage

A background picture can provide a watermark effect, enhancing a DataWindow's online and printed appearance.

Depending on the *execute* argument, the function either adds the background to the DataWindow or returns syntax that you use in a Modify function to add the background picture.

Use Syntax 1 to add a background picture at its exact size.

Examples

This example uses the of_SetBackground function to add the confid.bmp background picture to dw_employee, sizing it to fit:

```
String   ls_return

ls_Return = &
   dw_employee.inv_report.of_SetBackground &
      ("confid.bmp", TRUE, 20, 20, 0, 0, TRUE)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify error",ls_Return)
END IF
```

of_SetBorder

Description

Changes the border for one or more objects in a DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_SetBorder** (*border* {, *objectname* {, *band* {, *execute* } } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
border	Border enumerated data type specifying the border style: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised! Specify the default border style using the of_SetDefaultBorder function
objectname (optional)	String specifying the name of the object to change. If you specify an asterisk (*), the function changes the border for all objects in the band The default is an asterisk (*)

Argument	Description
band (optional)	String specifying the band in which to change the objects. The function ignores this argument if you pass a name in *objectname*. Valid values are: An asterisk (*), which changes all objects in the DataWindow Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber* The default is an asterisk (*)
execute (optional)	Boolean specifying whether to execute the Modify function that sets the border (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that changed the border style; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to change the border style.

Usage

The of_PrepPrint function calls this function to remove all border styles. You can use it to selectively change border styles before printing (for example, you might change 3d Lowered to Box).

Depending on the *execute* argument, the function either changes the border style or returns syntax that you use in a Modify function to change the border style.

Examples

This example changes the border for all objects in the header band to a box:

```
String    ls_return
ls_return = dw_employee.inv_report.of_SetBorder &
   (Box!, "*", "header", TRUE)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify error",ls_Return)
END IF
```

of_SetColor

Description

Changes the color and background color (if applicable) of one or more objects in a DataWindow.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetColor** (*color, backgroundcolor*
{, *objectname* {, *band* {, *execute* } } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
color	Long specifying a long value that identifies the new color. If you pass a NULL, the function does not change the color
backgroundcolor	Long specifying a long value that identifies the new background color. If you pass a NULL, the function does not change the background color
objectname (optional)	String specifying the name of the object to change. If you specify an asterisk (*), the function changes the border for all objects in the band The default is an asterisk (*)
band (optional)	String specifying the band in which to change the objects. The function ignores this argument if you pass a name in *objectname*. Valid bands are: An asterisk (*), which changes colors for all objects Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber* The default is an asterisk (*)
execute (optional)	Boolean specifying whether to execute the Modify function that sets the color (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that changed the color; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to change the color.

Usage

The of_PrepPrint function calls this function to change all text to black on white. You can use it to selectively change colors before printing.

Depending on the *execute* argument, the function either changes the color or returns syntax that you use in a Modify function to change the color.

Examples

This example changes the text color to black and the background color to white:

```
String    ls_return
Long      ll_white = (RGB(255, 255, 255))
Long      ll_black = 0

// Change all colors to black on white
ls_return = dw_employee.inv_report.of_SetColor &
   (ll_black, ll_white, "*", "*", TRUE)
IF Trim(ls_return) <> "" THEN
   MessageBox("Modify error",ls_Return)
END IF
```

of_SetDefaultBackColor

Description

Specifies the default background color.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultBackColor** (*color*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
color	Long specifying the default background color. If you pass a NULL, the function sets the background color to transparent

Return value

None

Examples

This example calls the of_SetDefaultBackColor function, setting the background color to transparent:

```
Long   ll_backcolor

SetNull(ll_backcolor)
dw_emp.inv_report.of_SetDefaultBackColor &
   (ll_backcolor)
```

of_SetDefaultBorder

Description

Specifies the default border.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultBorder** (*border*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
border	Border enumerated data type specifying the default border: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised!

Return value

None

Examples

This example calls the of_SetDefaultBorder function:

```
dw_emp.inv_report.of_SetDefaultBorder(Lowered!)
```

of_SetDefaultCharset

Description

Specifies the default character set.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultCharset** (*charset*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Argument	Description
charset	FontCharSet enumerated data type specifying the default character set: ANSI! Symbol! ChineseBig5! Hangul! ShiftJIS! OEM!

Return value

None

Examples

This example calls the of_SetDefaultCharset function:

```
dw_emp.inv_report.of_SetDefaultCharset(ANSI!)
```

of_SetDefaultColor

Description

Specifies the default color.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultColor** (*color*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
color	Long specifying the default color. If you pass a NULL, the function sets the color to transparent

Return value

None

Examples

This example calls the of_SetDefaultColor function:

```
Long   ll_color

ll_color = &
   dw_emp.inv_report.of_SetDefaultColor &
     (RGB(128, 128, 128))
```

of_SetDefaultFontFace

Description

Specifies the default font (for example, Arial).

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultFontFace** (*font*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
font	String specifying the default font

Return value

None

Examples

This example calls the of_SetDefaultFontFace function:

```
dw_emp.inv_report.of_SetDefaultFontFace("Arial")
```

of_SetDefaultFontSize

Description

Specifies the default font point size.

Access

Public

Syntax

dwcontrol.instancename.**of_SetDefaultFontSize** (*fontsize*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
fontsize	Integer specifying the default font size, in points

Return value

None

Examples

This example calls the of_SetDefaultFontSize function:

```
dw_emp.inv_report.of_SetDefaultFontSize(12)
```

of_SetFont

Changes the font of one or more objects in a DataWindow. There are two syntaxes:

To	Use
Set the font, including bold, italic, and underline	Syntax 1
Set the font, using defaults for bold, italic, and underline	Syntax 2

Syntax 1

Set the font, including bold, italic, and underline

Description

Sets the font, including bold, italic, and underline.

Access

Public

Syntax

dwcontrol.instancename.**of_SetFont** (*fontface*, *fontsize*, *fontfamily*, *fontpitch* {, *bold*, *italic*, *underline* {, *objectname* {, *band* {, *execute* } } } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
fontface	String specifying the new font. If you pass a NULL, the function does not change the font
fontsize	Integer specifying the point size of the new font. If you pass a NULL, the function does not change the font size
fontfamily	FontFamily enumerated data type specifying the font family: AnyFont! Roman! Swiss! Modern! Script! Decorative! If you pass a NULL, the function does not change the font family
fontpitch	FontPitch enumerated data type specifying the pitch of the new font: Default! Fixed! Variable! If you pass a NULL, the function does not change the pitch

Argument	Description
bold (optional)	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default) If you pass a NULL, the function does not change the bold property
italic (optional)	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default) If you pass a NULL, the function does not change the italic property
underline (optional)	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default) If you pass a NULL, the function does not change the underline property
objectname (optional)	String specifying the name of the object to change. If you specify an asterisk (*), the function changes the font for all objects in the band The default is an asterisk (*)
band (optional)	String specifying the band in which to change the objects. The function ignores this argument if you pass a name in *objectname*. Valid bands are: An asterisk (*), which changes the border for all objects Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber* The default is an asterisk (*)
execute (optional)	Boolean specifying whether to execute the Modify function that sets the font (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that changes the font; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to change the font.

Usage

Call this function to change the font for a DataWindow column before printing.

Depending on the *execute* argument, the function either changes the font or returns syntax that you use in a Modify function to change the font.

Use Syntax 2 to call this function, leaving bold, italic, and underline properties as is.

Examples

This example changes all text to Arial bold, using the default pitch. No other properties are changed:

```
String    ls_return
Integer   li_size
Boolean   lb_bold, lb_italic, lb_underline

SetNull(li_size)
SetNull(lb_italic)
SetNull(lb_underline)
lb_bold = TRUE
ls_return = dw_employee.inv_report.of_SetFont &
   ("Arial", li_size, Swiss!, Default!, &
      lb_bold, lb_italic, lb_underline, &
         "*", "*", TRUE)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
END IF
```

Syntax 2

Set the font, using defaults for bold, italic, and underline

Description

Sets the font, using defaults for bold, italic, and underline. This syntax always executes the Modify statement that changes the font.

Access

Public

Syntax

dwcontrol.instancename.**of_SetFont** (*fontface*, *fontsize*, *fontfamily*, *fontpitch* {, *objectname* {, *band* } })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
fontface	String specifying the new font. If you pass a NULL, the function does not change the font

Argument	Description
fontsize	Integer specifying the point size of the new font. If you pass a NULL, the function does not change the font size
fontfamily	FontFamily enumerated data type specifying the font family: AnyFont! Roman! Swiss! Modern! Script! Decorative! If you pass a NULL, the function does not change the font family
fontpitch	FontPitch enumerated data type specifying the pitch of the new font: Default! Fixed! Variable! If you pass a NULL, the function does not change the pitch
objectname (optional)	String specifying the name of the object to change. If you specify an asterisk (*), the function changes the font for all objects in the band The default is an asterisk (*)
band (optional)	String specifying the band in which to change the objects. The function ignores this argument if you pass a name in *objectname*. Valid bands are: An asterisk (*), which changes the border for all objects Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber* The default is an asterisk (*)

Return value

String. Returns the result of the Modify statement that changes the font; an empty string indicates success.

Usage

Call this function to change the font for a DataWindow column before printing.

Use Syntax 1 to call this function, with the addition of bold, italic, and underline properties.

Examples

This example changes all text to Arial, using the default pitch. No other properties are changed:

```
String    ls_return
Integer   li_size

SetNull(li_size)
ls_return = dw_employee.inv_report.of_SetFont &
   ("Arial", li_size, Swiss!, Default!, &
      "*", "*", TRUE)
```

of_SetRelativeZoom

Description

Increases or decreases the size of a report relative to its current size.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRelativeZoom** (*zoompercentage*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
zoompercentage	Integer specifying the percentage change to the DataWindow (50% = half the size, 200% = twice the size, 100% = no change)

Return value

String. Returns the output of the Modify command. An empty string ("") indicates success.

Usage

This change is relative to the current report's zoom percentage. To change the size to an absolute percentage, code a Modify statement that specifies DataWindow.Zoom.

Examples

This example reduces the dw_employee DataWindow's size by 50 percent:

```
String    ls_return

ls_return = &
   dw_employee.inv_report.of_SetRelativeZoom(50)
IF ls_Return <> "" THEN
   MessageBox("Print Error", &
      "DataWindow.Zoom error")
```

```
END IF
```

of_ShiftBand

Description

Shifts the objects in the given band of the DataWindow horizontally or vertically by a specified number of units.

Access

Public

Syntax

dwcontrol.instancename.**of_ShiftBand** (*band, xunits, yunits* {, *execute* })

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
band	String specifying the band to shift. Valid bands are: An asterisk (*), which shifts entire the DataWindow Header Detail Footer Summary Header.*levelnumber* Trailer.*levelnumber*
xunits	Integer specifying the number of units to shift the objects right (to shift left, pass a negative number)
yunits	Integer specifying the number of units to shift the objects down (to shift up, pass a negative number)
execute (optional)	Boolean specifying whether to execute the Modify function that shifts the band (TRUE) or return a string that you use to execute the Modify function (FALSE). The default is TRUE

Return value

String. If *execute* is TRUE, this function returns the result of the Modify statement that shifted the objects; an empty string indicates success. If *execute* is FALSE, this function returns a string that you use as input to a Modify function to shift the objects.

Usage

Call this function to reposition existing objects when adding other objects dynamically. For example, if you add a bitmap to a band, call the of_GetPictureSize function to determine the bitmap's size and then call this function to reposition the other objects.

Depending on the *execute* argument, the function either shifts the objects or returns syntax that you use in a Modify function to shift the objects.

Examples

This example calls the of_ShiftBand function:

```
String    ls_return

ls_return = dw_employee.inv_report.of_ShiftBand &
   ("header", 50, 0, TRUE)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
END IF
```

of_Undo

Description

Restores changes made to a DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_Undo** (*levelstoundo*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
levelstoundo	Integer specifying the number of changes to restore. Specify 0 to undo all changes

Return value

String. Returns the output of the Modify command. An empty string ("") indicates success.

Usage

Call this function to undo changes made while viewing an online report. This function is used internally by of_PrintReport to undo the changes made by of_PrepPrint.

If you are buffering multiple strings to use in a single Modify function, each function call counts as a separate undo level. For example, if you might call of_AddCompute, of_AddLine, and of_AddText, with *execute* set to FALSE, for input to a single Modify function. To remove these changes, call of_Undo(3).

This function is for undoing cosmetic changes made while preparing a DataWindow for online reporting and hardcopy printing. To undo changes made to DataWindow data, use the n_cst_dwsrv_rowmanager of_Undelete function.

Examples

This example uses the of_Undo function after printing a DataWindow:

```
String    ls_return

// Prepare for printing.
ls_Return = &
   dw_employee.inv_report.of_PrepPrint(TRUE)
IF ls_Return <> "" THEN
   MessageBox("Print Error", &
      "Error preparing DataWindow for printing")
END IF

// Print report.
dw_employee.Print(TRUE)

// Undo changes made by of_PrepPrint.
ls_return = dw_employee.inv_report.of_Undo(1)
IF Trim(ls_Return) <> "" THEN
   MessageBox("Modify Error",ls_Return)
END IF
```

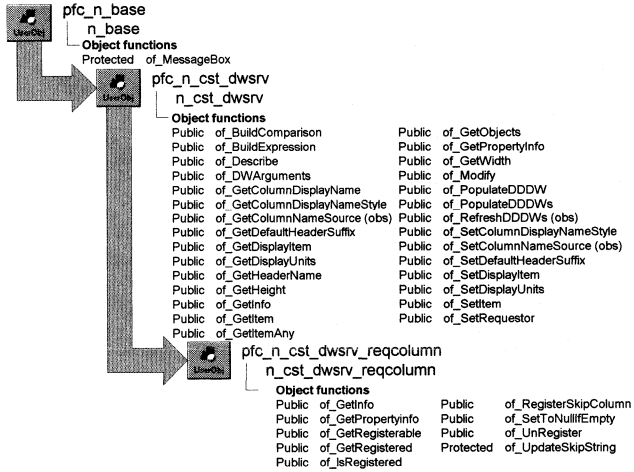
n_cst_dwsrv_reqcolumn

Description

DataWindow required column service. This service enables and disables default DataWindow processing for required fields, making it easier for your application to handle interdependent fields within a DataWindow.

This service applies only to DataWindow columns that use the Nilisnull property. For example, EditMasks don't have this property so the required column service doesn't apply to edit masks.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

This service allows you to defer required fields processing until the user completes data entry. It also allows you to specify columns for which PowerBuilder should still perform required fields processing.

When enabled, this service writes NULL values to empty fields with the Nilisnull property, suppressing required fields messages.

To use the required column service:

1 Enable the service using the u_dw of_SetReqColumn function:

```
dw_emp.of_SetReqColumn(TRUE)
```

2 Call the of_RegisterSkipColumn to selectively enable default DataWindow required fields processing:

```
dw_emp.inv_reqcolumn.of_RegisterSkipColumn &
   ("dept_id")
```

See also

u_dw

Instance variables

Instance variable	Description	Data type	Access	Usage
is_skip	Tracks skipped columns	String	Protected	Internal
is_skipcolumn[ ]	Specifies columns for which default DataWindow required fields processing is enabled	String	Protected	Internal

Functions

N_cst_dwsrv_reqcolumn includes precoded object functions:

of_GetInfo	of_RegisterSkipColumn
of_GetPropertyInfo	of_SetToNullIfEmpty
of_GetRegisterable	of_UnRegister
of_GetRegistered	of_UpdateSkipString
of_IsRegistered	

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_reqcolumn.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop
```

```
dw_1.inv_reqcolumn.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegisterable

Description

Retrieves an array of registerable columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterable** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
columns	String array into which the function places a list of registerable columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Examples

This example calls the of_GetRegisterable function:

```
String   ls_columns[ ]
Integer  li_count, li_loop

li_loop =  &
   dw_1.inv_reqcolumn.of_GetRegisterable  &
     (ls_columns)
...
```

of_GetRegistered

Description

Retrieves an array of registered columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegistered** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
columns	String array into which the function places a list of registered columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Examples

This example calls the of_GetRegistered function:

```
String   ls_columns[ ]
Integer  li_count, li_loop

li_loop =  &
   dw_1.inv_reqcolumn.of_GetRegistered  &
     (ls_columns)
...
```

of_IsRegistered

Description

Reports whether a column is registered.

Access

Public

Syntax

dwcontrol.instancename.**of_IsRegistered** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
column	String specifying the column to be checked

Return value

Boolean. Returns TRUE if *column* is registered and FALSE if it is not.

Examples

This example calls the of_IsRegistered function:

```
IF NOT dw_1.inv_reqcolumn.of_IsRegistered  &
  ("termination_date") THEN
    dw_1.inv_reqcolumn.of_RegisterSkipColumn  &
      ("termination_date")
END IF
```

of_RegisterSkipColumn

Description

Specifies that the specified column uses default DataWindow required field processing.

Access

Public

Syntax

dwcontrol.instancename.**of_RegisterSkipColumn** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
column	String specifying the column for which default DataWindow required fields processing applies

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_RegisterSkipColumn function:

```
Integer   li_return

li_return = &
   dw_emp.inv_reqcolumn.of_RegisterSkipColumn &
      ("emp_id")
```

of_SetToNullIfEmpty

Description

Replaces empty strings with a null value, suppressing the DataWindow's Value Required dialog box.

Access

Public

Syntax

dwcontrol.instancename.**of_SetToNullIfEmpty** (*row*, *column*, *text*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
row	Long specifying the row to check
column	String specifying the column to check
text	String specifying the current text

Return value

Integer. Returns values as follows:

- **3** The function rejects the data value but allows focus to change; the current value is NULL
- **1** The function converts the empty value to NULL
- **0** No action taken
- **-1** An error occurred

Usage

The u_dw ItemError event calls this function.

Examples

This example is from the u_dw ItemError event:

```
w_master   lw_parent

IF IsValid(inv_reqcolumn) THEN
   IF inv_reqcolumn.of_SetToNullIfEmpty &
      (row, string(dwo.Name), &
         this.GetText()) > 0 THEN
            Return 3
   END IF
END IF
...
```

of_UnRegister

Description

Removes a column from the is_skipcolumn array.

Access

Public

Syntax

dwcontrol.instancename.**of_UnRegister** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
column	String specifying the column to be removed from the is_skipcolumn array

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Once removed from the array, default DataWindow required fields processing is enabled.

Examples

This example calls the of_UnRegister function:

```
Integer   li_return

li_return = &
   dw_emp.inv_reqcolumn.of_UnRegister &
      ("emp_id")
```

of_UpdateSkipString

Description

Creates a skip string.

Access

Protected

Syntax

dwcontrol.instancename.**of_UpdateSkipString** (*skipcolumns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
skipcolumns	String array used by the function to construct the skip string

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_RegisterSkipColumn function:

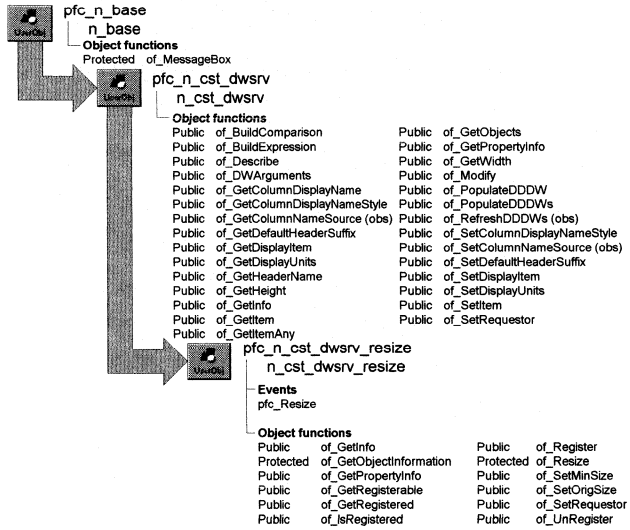
```
...
of_UpdateSkipString(is_skipcolumn)
...
```

n_cst_dwsrv_resize

Description

DataWindow resize service. This object resizes the objects within a DataWindow object, including columns, computed fields, bitmaps, and text.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

Use this object's functions to enable resizing of the objects displayed in a DataWindow object (so that when the user resizes the window, this service resizes DataWindow contents automatically).

Presentation styles
You cannot use the DataWindow resize service with DataWindow objects that have the Composite or RichTextEdit presentation style.

To use the DataWindow resize service:

1 Enable the DataWindow resize service using the u_dw of_SetResize function (this example is from a DataWindow Constructor event):

```
this.of_SetResize(TRUE)
```

2 (Optional) Call the of_SetMinSize function to specify a minimum size below which the DataWindow resize service no longer resizes DataWindow contents:

```
this.inv_resize.of_SetMinSize &
(this.width-50, this.height-100)
```

3 Specify the objects to be resized and how they should be resized by calling the of_Register function:

```
this.inv_resize.of_Register("emp_fname", &
0, 0, 50, 50)
this.inv_resize.of_Register("emp_lname", &
100, 0, 50, 50)
```

4 Enable the window resize service and register the DataWindow control (this example is from a window Open event):

```
this.of_SetResize(TRUE)
this.inv_resize.of_Register(dw_1, 0, 0, 100, 100)
```

5 (Optional) Call the of_UnRegister function to remove objects from the resize list.

Structures

Structure	Field	Data type
os_resize	b_movex	Boolean
	b_movey	Boolean
	b_scale	Boolean
	b_scaleheight	Boolean
	b_scalewidth	Boolean
	i_movex	Integer
	i_movey	Integer
	i_scaleheight	Integer
	i_scalewidth	Integer
	r_height	Real
	r_width	Real
	r_x	Real
	r_y	Real

Structure	Field	Data type
	s_control	String
	s_typeof	String

See also

n_cst_dwsrv
n_cst_resize
u_dw

Instance variables

N_cst_dwsrv_resize includes instance variables:

Instance variable	Description	Data type	Access	Usage
EMPTY	Constant set to an empty string	String	Protected	Internal
FIXEDBOTTOM	Constant set to FixedToBottom	String	Public	Use with of_Register
FIXEDBOTTOM_SCALERIGHT	Constant set to FixedToBottom&ScaleToRight	String	Public	Use with of_Register
FIXEDRIGHT	Constant set to FixedToRight	String	Public	Use with of_Register
FIXEDRIGHT_SCALEBOTTOM	Constant set to FixedToRight&ScaleToBottom	String	Public	Use with of_Register
FIXEDRIGHTBOTTOM	Constant set to FixedToRight&Bottom	String	Public	Use with of_Register
ii_rounding	Rounding factor	Integer	Protected	Internal (default is 5)
il_parentminimumheight	Minimum height of DataWindow control	Long	Protected	Set with of_SetMinSize
il_parentminimumwidth	Minimum width of DataWindow control	Long	Protected	Set with of_SetMinSize
il_parentprevheight	Original height of DataWindow control	Long	Protected	Set with of_SetOrigSize
il_parentprevwidth	Original width of DataWindow control	Long	Protected	Set with of_SetOrigSize

Instance variable	Description	Data type	Access	Usage
istr_registered[]	Array of registered columns and text objects	os_resize	Protected	Set with of_Register
LINE	Constant set to "line"	String	Protected	Internal
SCALE	Constant set to "Scale"	String	Public	Use with of_Register
SCALEBOTTOM	Constant set to "ScaleToBottom"	String	Public	Use with of_Register
SCALERIGHT	Constant set to "ScaleToRight"	String	Public	Use with of_Register
SCALERIGHTBOTTOM	Constant set to "ScaleToRight&Bottom"	String	Public	Use with of_Register

Events

N_cst_dwsrv_resize includes a precoded event:

pfc_Resize

pfc_Resize

Description

Calls the of_Resize function.

Syntax

dwcontrol.instancename.EVENT **pfc_Resize** (*sizetype*, *newwidth*, *newheight*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)

Argument	Description
sizetype	UnsignedLong specifying the resize type: ♦ 0 — (Restored) The window or DataWindow has been resized, but it was not minimized or maximized. The user may have dragged the borders or a script may have called the Resize PowerScript function ♦ 1 — (Minimized) The window or DataWindow has been minimized ♦ 2 — (Maximized) The window or DataWindow has been maximized
newwidth	Integer specifying the new DataWindow width
newheight	Integer specifying the new DataWindow height

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

This event is called by the u_dw Resize event.

Functions

N_cst_dwsrv_resize contains precoded object functions:

of_GetInfo	of_Register
of_GetObjectInformation	of_Resize
of_GetPropertyInfo	of_SetMinSize
of_GetRegisterable	of_SetOrigSize
of_GetRegistered	of_SetRequestor
of_IsRegistered	of_UnRegister

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control

Argument	Description
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_resize.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetObjectInformation

Description

Retrieves information on a specified object within a DataWindow.

Access

Protected

Syntax

dwcontrol.instancename.**of_GetObjectInformation** (*control*, *type*, *x*, *y*, *width*, *height*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
control	String specifying the object for which the function returns information
type	String into which the function places the object type Object types include column text, bitmap, compute, and so on (passed by reference)
x	Integer into which the function places the x coordinate (passed by reference)

Argument	Description
y	Integer into which the function places the y coordinate (passed by reference)
width	Integer into which the function places the object's width (passed by reference)
height	Integer into which the function places the object's height (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Resize function:

```
...
IF Len(istr_registered[li_cnt].s_control) > 0 THEN
  li_rc = of_GetObjectInformation &
    (istr_registered[li_cnt].s_control, &
      ls_type, li_x, li_y, li_width, &
        li_height)
...
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_resize.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegisterable

Description

Retrieves an array of registerable columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterable** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
columns	String array into which the function places a list of registerable columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Examples

This example calls the of_GetRegisterable function:

```
String   ls_columns[ ]
Integer  li_count, li_loop
li_loop =  &
   dw_1.inv_resize.of_GetRegisterable  &
     (ls_columns)
lb_1.Reset()
IF li_loop <> 0 THEN
  FOR li_count = 1 to li_loop
    lb_1.AddItem(ls_columns[li_count])
  NEXT
ELSE
  lb_1.AddItem("No Registerable columns")
END IF
```

of_GetRegistered

Description

Retrieves information for a registered object.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegistered** (*object*, *scale*, *movex*, *movey*, *scalewidth*, *scaleheight*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	DWObject or string specifying the object for which information is returned
scale	Boolean into which the function places the object's scale switch (passed by reference)
movex	Integer into which the function places the object's move x specification (passed by reference)
movey	Integer into which the function places the object's move y specification (passed by reference)
scalewidth	Integer into which the function places the object's scale width specification (passed by reference)
scaleheight	Integer into which the function places the object's scale height specification (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetRegistered function:

```
...
dw_1.inv_resize.of_GetRegistered  &
    ("dept_id", lb_scale, li_x, li_y,  &
       li_scalewidth, li_scaleheight)
...
```

of_IsRegistered

Description

Reports whether an object is registered.

Access

Public

Syntax

dwcontrol.instancename.**of_IsRegistered** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
column	String specifying the column to be checked

Return value

Boolean. Returns TRUE if *column* is registered and FALSE if it is not.

Examples

This example calls the of_IsRegistered function:

```
IF NOT dw_1.inv_resize.of_IsRegistered  &
  ("dept_id") THEN
    dw_1.inv_resize.of_Register("dept_id",  &
      0, 0, 50, 50)
END IF
```

of_Register

Registers an object within a DataWindow. The DataWindow resize service acts on registered objects only. There are four syntaxes:

To register by specifying a	Use
DWObject variable and resize percentage	Syntax 1
DWObject variable and resize method	Syntax 2
String and resize percentage	Syntax 3
String and resize method	Syntax 4

One additional syntax

N_cst_dwsrv_resize supports one additional of_Register syntax, which is defined as protected and used internally by the other syntaxes.

For more information on this syntax, use the Object Browser.

Syntax 1

Specify a DWObject variable and resize percentages

Description

Registers an object for resizing, given a DWObject variable and a set of resize percentages.

Access

Public

Syntax

dwcontrol.instancename.**of_Register** (*object*, *movex*, *movey*, *scalewidth*, *scaleheight*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	DWObject variable specifying the object to be resized
movex	Integer specifying the percentage to move the object along the x axis
movey	Integer specifying the percentage to move the object along the y axis
scalewidth	Integer specifying the percentage to resize the object along the x axis
scaleheight	Integer specifying the percentage to resize the object along the y axis

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You typically use either this syntax or Syntax 3 to register objects with the DataWindow resize service.

Examples

This example calls the of_Register function:

```
DWObject ldwo_obj

ldwo_obj = this.Object.emp_lname
this.inv_resize.of_Register &
(ldwo_obj, inv_resize.0, 0, 50, 50)
ldwo_obj = this.Object.emp_fname
this.inv_resize.of_Register &
 (ldwo_obj, 100, 0, 50, 50)
this.of_SetTransObject(SQLCA)
this.pfc_Retrieve()
```

Syntax 2

Specify a DWObject variable and resize method

Description

Registers an object for resizing, given a DWObject variable and a resize method.

Access

Public

Syntax

*dwcontrol.instancename.***of_Register** (*object, resizemethod*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	DWObject variable specifying the object to be resized
resizemethod	String or n_cst_dwsrv_resize constant specifying the action the service takes when the DataWindow Resize event is triggered: ♦ "FixedToRight" or FIXEDRIGHT ♦ "FixedToBottom" or FIXEDBOTTOM ♦ "FixedToRight&Bottom" or FIXEDRIGHTBOTTOM ♦ "Scale" or SCALE ♦ "ScaleToRight" or SCALERIGHT ♦ "ScaleToBottom" or SCALEBOTTOM ♦ "ScaleToRight&Bottom" or SCALERIGHTBOTTOM ♦ "FixedToRight&ScaleToBottom" or FIXEDRIGHT_SCALEBOTTOM ♦ "FixedToBottom&ScaleToRight" or FIXEDBOTTOM_SCALERIGHT

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Because explicit move and resize percentages provide more flexibility when handling multiple objects, it's usually best to use Syntax 1 instead.

Examples

This example calls the of_Register function:

```
DWObject ldwo_obj

ldwo_obj = this.Object.comments
this.inv_resize.of_Register &
  (ldwo_obj, inv_resize.SCALEBOTTOM)
this.of_SetTransObject(SQLCA)
this.pfc_Retrieve()
```

Syntax 3 **Specify a string and resize percentages**

Description

Registers an object for resizing, given a string identifying the object and a set of resize percentages.

Access

Public

Syntax

dwcontrol.instancename.**of_Register** (*object, movex, movey, scalewidth, scaleheight*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	String specifying the object to be resized
movex	Integer specifying the percentage to move the object along the x axis
movey	Integer specifying the percentage to move the object along the y axis
scalewidth	Integer specifying the percentage to resize the object along the x axis
scaleheight	Integer specifying the percentage to resize the object along the y axis

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You typically use either this syntax or Syntax 1 to register objects with the DataWindow resize service.

Examples

This example calls the of_Register function:

```
this.of_SetResize(TRUE)
this.inv_resize.of_Register &
  ("emp_fname", 0, 0, 50, 50)
this.inv_resize.of_Register &
  ("emp_lname", 100, 0, 50, 50)
this.Event pfc_Retrieve()
```

Syntax 4 **Specify a string and resize method**

Description

Registers an object for resizing, given a string identifying the object and a resize method.

Access

Public

Syntax

*dwcontrol.instancename.***of_Register** (*object, resizemethod*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	String specifying the object to be resized
resizemethod	String or n_cst_dwsrv_resize constant specifying the action the service takes when the DataWindow Resize event is triggered: ♦ "FixedToRight" or FIXEDRIGHT ♦ "FixedToBottom" or FIXEDBOTTOM ♦ "FixedToRight&Bottom" or FIXEDRIGHTBOTTOM ♦ "Scale" or SCALE ♦ "ScaleToRight" or SCALERIGHT ♦ "ScaleToBottom" or SCALEBOTTOM ♦ "ScaleToRight&Bottom" or SCALERIGHTBOTTOM ♦ "FixedToRight&ScaleToBottom" or FIXEDRIGHT_SCALEBOTTOM ♦ "FixedToBottom&ScaleToRight" or FIXEDBOTTOM_SCALERIGHT

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Because explicit move and resize percentages provide more flexibility when handling multiple objects, it's usually best to use Syntax 3 instead.

Examples

This example calls the of_Register function:

```
this.inv_resize.of_Register &
  ("comments", inv_resize.SCALEBOTTOM)
this.of_SetTransObject(SQLCA)
this.pfc_Retrieve()
```

of_Resize

Description

Moves or resizes registered objects as specified in the of_Register function.

Access

Protected

Syntax

dwcontrol.instancename.**of_Resize** (*newwidth*, *newheight*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
newwidth	Integer specifying the new DataWindow width
newheight	Integer specifying the new DataWindow height

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The pfc_Resize event calls this function when the DataWindow is resized.

Examples

This example is from the pfc_Resize event:

```
Return of_Resize(ai_newwidth, ai_newheight)
```

of_SetMinSize

Description

Sets a minimum DataWindow size below which the resize service no longer modifies the position of registered objects.

Access

Public

Syntax

dwcontrol.instancename.**of_SetMinSize** (*minimumwidth*, *minimumheight*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
minimumwidth	Integer specifying minimum width
minimumheight	Integer specifying minimum height

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Test different control sizes in the Window or User Object painter to determine an appropriate minimum size.

Examples

This example calls the of_SetMinSize function in a DataWindow Constructor event:

```
this.of_SetResize(TRUE)
```

```
this.inv_resize.of_SetMinSize &
  (this.width - 50, this.height - 100)
...
```

of_SetOrigSize

Description

Saves the current DataWindow control size.

Access

Public

Syntax

dwcontrol.instancename.**of_SetOrigSize** (*width, height*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
width	Integer specifying the DataWindow width
height	Integer specifying the DataWindow height

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The u_dw of_SetResize function calls this function to set the original DataWindow control size. If you want to modify the original size, you must call this function before calling the of_Register function.

Examples

This example is from the u_dw of_SetResize function:

```
IF ab_switch THEN
  IF IsNull(inv_resize) OR NOT &
    IsValid(inv_resize) THEN
      inv_resize = CREATE n_cst_dwsrv_resize
      inv_resize.of_SetRequestor(this)
      inv_resize.of_SetOrigSize &
        (this.Width, this.Height)
      Return 1
  END IF
...
```

of_SetRequestor

Description

Associates a DataWindow control with this instance of n_cst_dwsrv_resize and ensures that the DataWindow uses a valid presentation style. The DataWindow resize service does not support the Composite and RichTextEdit presentation styles.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRequestor** (*requestor*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
requestor	U_dw-based DataWindow window control to associate with this instance of n_cst_dwsrv_resize

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The u_dw of_SetResize function calls this function.

Examples

This example is from the u_dw of_SetResize function:

```
IF ab_Switch THEN
  IF IsNull(inv_resize) OR NOT &
    IsValid(inv_resize) THEN
      inv_resize = CREATE n_cst_dwsrv_resize
      inv_resize.of_SetRequestor(this)
      inv_resize.of_SetOrigSize &
        (this.Width, this.Height)
      Return 1
  END IF
...
```

of_UnRegister

Removes an object from the list of registered objects. There are two syntaxes:

To unregister by specifying a	Use
DWObject variable	Syntax 1
String	Syntax 2

Syntax 1 **Specify a DWObject**

Description

Removes an object from the list of registered objects, given a DWObject variable.

Access

Public

Syntax

dwcontrol.instancename.**of_UnRegister** (*object*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	DWObject variable specifying the object to be removed from the list of registered objects

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_UnRegister function:

```
DWObject ldwo_obj

ldwo_obj = this.Object.emp_lname
this.inv_resize.of_UnRegister (ldwo_obj)
```

Syntax 2 **Specify a string**

Description

Removes an object from the list of registered objects, given a string.

Access

Public

Syntax

dwcontrol.instancename.**of_UnRegister** (*object*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
object	String specifying the object to be removed from the list of registered objects

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_UnRegister function:

```
this.inv_resize.of_UnRegister ("emp_fname")
```

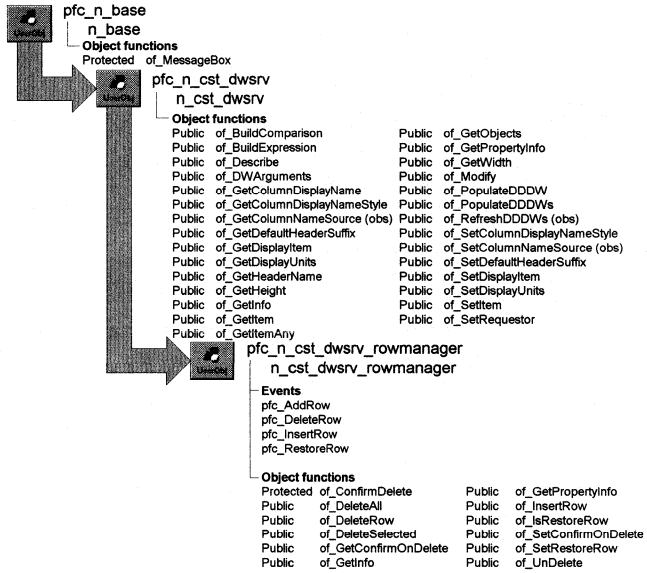
n_cst_dwsrv_rowmanager

Description

Row management service. Contains functions and events that allow you to:

- Add an empty row to the end of the DataWindow
- Insert an empty row between two existing rows
- Delete one or more rows
- Restores the last delete

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

Use this service to enhance your application's row management capabilities. You can insert rows, delete rows, and restore deleted rows.

To use this service:

1 Enable the service using the u_dw of_SetRowManager function:

```
dw_emp.of_SetRowManager(TRUE)
```

2 (Optional) Enable restore row functionality by calling the of_SetRestoreRow function:

```
dw_emp.inv_rowmanager.of_SetRestoreRow(TRUE)
```

3 Call row management events and functions as needed.

See also

n_cst_dwsrv_linkage
u_dw

Instance variables

N_cst_dwsrv_rowmanager includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_confirmondelete	Indicates whether the service prompts users before deleting rows	Boolean	Protected	Internal
ib_restorerow	Indicates whether restore row functionality is enabled	Boolean	Protected	Set with of_SetRestoreRow

Events

N_cst_dwsrv_rowmanager includes precoded events:

pfc_AddRow	pfc_InsertRow
pfc_DeleteRow	pfc_RestoreRow

pfc_AddRow

Description

Inserts a new row at the end of the DataWindow and sets focus.

Return value

Long. Returns the number of the inserted row if the event succeeds and -1 if an error occurs.

Usage

This event executes when the user selects Add from the m_dw popup menu.

pfc_DeleteRow

Description

Deletes the currently selected row or rows.

Return value

Long. Returns the number of rows deleted if the event succeeds and -1 if an error occurs.

Usage

This event executes when the user selects Delete from the m_dw popup menu.

pfc_InsertRow

Description

Inserts a new row into the associated DataWindow and sets focus.

Syntax

dwcontrol.instancename.Event **pfc_InsertRow** (*nextrow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager).
nextrow	Long specifying the row before which the event adds a row. From within the pfc_InsertRow event, access this value through the *al_beforerow* argument

Return value

Long. Returns the number of the inserted row if the event succeeds and -1 if an error occurs.

Usage

This event executes when the user selects Insert from the m_dw popup menu.

pfc_RestoreRow

Description

Displays the w_restorerow dialog box, allowing users to restore previously deleted rows.

Return value

Long. Returns the number of rows retrieved if the event succeeds, -1 if an error occurs, and -2 if there are no rows in the undelete buffer.

Usage

This event executes when the user selects Restore from the m_dw popup menu.

Functions

N_cst_dwsrv_rowmanager includes precoded object functions:

of_ConfirmDelete	of_GetPropertyInfo
of_DeleteAll	of_InsertRow
of_DeleteRow	of_IsRestoreRow
of_DeleteSelected	of_SetConfirmOnDelete
of_GetConfirmOnDelete	of_SetRestoreRow
of_GetInfo	of_UnDelete

of_ConfirmDelete

Description

Displays a message box, prompting the user to confirm a deletion request.

Access

Protected

Syntax

dwcontrol.instancename.**of_ConfirmDelete** (*amount*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
amount	Long specifying the number of rows to be deleted

Return value

Integer. Returns 1 if the deletion should continue, 2 if the deletion should not continue, and -1 if an error occurs.

Usage

Internal.

Examples

This example calls the of_ConfirmDelete function:

```
...
IF ib_confirmondelete THEN
   li_confirm = of_ConfirmDelete(ll_rowcount)
END IF
...
```

of_DeleteAll

Description

Deletes all rows from the associated DataWindow's primary buffer.

Access

Public

Syntax

dwcontrol.instancename.**of_DeleteAll** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)

Return value

Integer. Returns the number of rows deleted if the function succeeds and -1 if an error occurs.

Usage

You must still code a PowerScript Update function to apply the changes to the database.

Examples

This example deletes all rows:

```
Integer   li_return

li_return = &
   dw_emplist.inv_rowmanager.of_DeleteAll( )
...
```

of_DeleteRow

Description

Deletes the specified row in the DataWindow. After the delete, the function sets focus to the next row in the DataWindow at the first column with a nonzero tab order.

Access

Public

Syntax

dwcontrol.instancename.**of_DeleteRow** (*rownumber*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
rownumber	Long specifying the row to be deleted. If *rownumber* is zero, the function deletes the current row

Return value

Integer. Returns 1 if the row is deleted successfully, 0 if the row was not deleted, and -1 if an error occurs.

Usage

You must still code a PowerScript Update function to apply the changes to the database.

You can call of_UnDelete to reverse deletions made by of_DeleteRow.

Examples

This example deletes the current row:

```
Integer   li_return

li_return = dw_emplist.inv_rowmanager.of_DeleteRow(0)
...
```

of_DeleteSelected

Description

Deletes all selected rows in the DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_DeleteSelected ( )**

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)

Return value

Integer. Returns the number of rows deleted if the function succeeds and -1 if an error occurs.

Usage

Call this function in conjunction with the row selection service to delete multiple rows in a DataWindow. You must still code an Update function to apply the changes to the database.

Examples

This example deletes all selected rows. It assumes that you have enabled multirow selection via the u_dw of_SetRowSelect function:

```
Integer   li_return

li_return = &
   dw_emplist.inv_rowmanager.of_DeleteSelected( )
...
```

of_GetConfirmOnDelete

Description

Reports whether the service prompts the user to confirm deletion requests.

Access

Public

Syntax

dwcontrol.instancename.**of_GetConfirmOnDelete** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)

Return value

Boolean. Returns TRUE if the service confirms deletions and FALSE if it does not.

Examples

This example calls the of_GetConfirmOnDelete function:

```
n_cst_conversion   lnv_conversion
Boolean   lb_confirm

lb_confirm = &
   dw_emp.inv_rowmanager.of_GetConfirmOnDelete( )
   MessageBox("Delete", "Confirm on delete is " &
      + lnv_conversion.of_String(lb_confirm) )
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info
```

```
dw_1.inv_rowmanager.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_rowmanager.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_InsertRow

Description

Inserts a row in the DataWindow at a specified place, then sets focus to the first column with tab order and scrolls to the new row.

Access

Public

Syntax

dwcontrol.instancename.**of_InsertRow** (*insertionpoint*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
insertionpoint	Long specifying the row before which the function inserts the new row. Specify 0 to insert at the end of all rows

Return value

Long. Returns the new row number if the function succeeds and -1 if an error occurs.

Examples

This example inserts a new row at the end of the DataWindow:

```
dw_emplist.inv_rowmanager.of_InsertRow(0)
```

of_IsRestoreRow

Description

Reports whether restore row functionality is enabled.

Access

Public

Syntax

dwcontrol.instancename.**of_IsRestoreRow** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)

Return value

Boolean. Returns TRUE if restore row functionality is enabled and FALSE if it is not.

Examples

This example calls the of_IsRestoreRow function:

```
IF dw_1.inv_rowmanager.of_IsRestoreRow() THEN
   MessageBox("RowMan", "Restore row is enabled")
ELSE
   MessageBox("RowMan", "Restore row is disabled")
END IF
```

of_SetConfirmOnDelete

Description

Specifies whether the service prompts the user to confirm deletion requests.

Access

Public

Syntax

dwcontrol.instancename.**of_SetConfirmOnDelete** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
boolean	Boolean indicating whether the service prompts the user to confirm deletion request (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetConfirmOnDelete function:

```
dw_emp.inv_rowmanager.of_SetConfirmOnDelete(FALSE)
```

of_SetRestoreRow

Description

Specifies whether restore row functionality is enabled.

Access

Public

Syntax

dwcontrol.instancename.**of_SetRestoreRow** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
boolean	Boolean indicating whether to enable (TRUE) or disable (FALSE) restore row functionality

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

When restore row functionality is enabled, users can display the w_restorerow dialog box by right clicking and selecting Restore from the popup menu.

Examples

This example calls the of_SetRestoreRow function:

```
dw_1.of_SetRowManager(TRUE)
```

```
dw_1.inv_rowmanager.of_SetRestoreRow(TRUE)
```

of_UnDelete

Description

Displays the w_restorerow dialog box, allowing users to restore deleted rows.

Access

Public

Syntax

dwcontrol.instancename.**of_UnDelete** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)

Return value

Long. Returns the number of restored rows if the function succeeds, -1 if an error occurs, and -2 if the delete buffer is empty.

Usage

Call this function to display the w_restorerow dialog box, which allows users to restore deleted rows from the Deleted buffer. This means that newly added rows that you delete before saving to the database can't be undeleted, since they never get copied to the Deleted buffer.

Examples

This example calls the of_UnDelete function:

```
IF dw_emplist.inv_rowmanager.of_UnDelete( ) = -1 THEN
   MessageBox("Undelete", "Undelete error")
ELSE
   gnv_app.of_GetFrame( ).SetMicroHelp &
      ("Undelete was successful")
END IF
```

n_cst_dwsrv_rowselection

Description

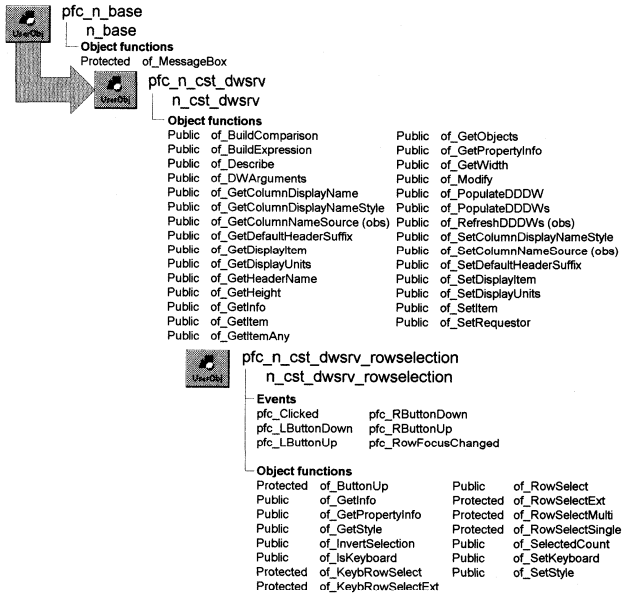
Row selection service. You can implement different styles of this service:

- **Single-row selection** Handles row selection when your DataWindow allows one row to be selected at a time.
- **Multirow selection** Handles row selection by allowing your runtime users to select multiple rows with single clicks. These rows can be contiguous or noncontiguous.

  When multirow selection is enabled, runtime users toggle a row's selected state by clicking it. This capability is similar to a ListBox's MultiSelect property.
- **Extended selection** Handles row selection by allowing your runtime users to select multiple rows with SHIFT+click, CTRL+click, and CTRL+SHIFT+click.

  This capability is similar to a ListBox's ExtendedSelect property.

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib

Usage

This service handles all row selection automatically. You don't need to add any code to implement row selection itself.

To use this service:

1 Enable the service using the u_dw of_SetRowSelect function:

```
dw_emp.of_SetRowSelect(TRUE)
```

2 Specify the row selection option by calling the of_SetStyle function with one of the following arguments:

- **SINGLE or 0** Single-row selection
- **MULTIPLE or 1** Multirow selection
- **EXTENDED or 2** Extended selection

This example enables extended selection:

```
dw_emp.inv_rowselect.of_SetStyle  &
   (dw_emp.inv_rowselect.EXTENDED)
```

3 Add logic to your application to access and process selected rows. For example, you might call the row manager service's pfc_DeleteRow event to delete all selected rows:

```
IF dw_emp.inv_rowmanager.Event pfc_DeleteRow() &
   = -1 THEN
      MessageBox("Error", "Deletion error")
END IF
```

See also

n_cst_dwsrv_rowmanager
n_cst_selection
u_dw

Instance variables

N_cst_dwsrv_rowselection includes instance variables:

Instance variable	Description	Data type	Access	Usage
EXTENDED	Constant set to 2	Integer	Public	Use with of_SetStyle
ib_keyboard	Indicates whether keyboard support is active	Boolean	Protected	Internal

Instance variable	Description	Data type	Access	Usage
ib_lbuttonpressed	Indicates that the user is pressing the left mouse button	Boolean	Protected	Internal
ib_prevcntrl	Indicates whether the user pressed the CTRL key the last time a row was clicked	Boolean	Protected	Internal
ib_prevshift	Indicates whether the user pressed the SHIFT key the last time a row was clicked	Boolean	Protected	Internal
ib_rbuttonpressed	Indicates that the user is pressing the right mouse button	Boolean	Protected	Internal
ii_style	Indicates whether single-row, multirow, or extended selection service is enabled	Integer	Protected	Internal
il_anchorrow	Number of the row that is the current anchor point	Long	Protected	Internal
il_currclickedrow	Tracks the number of the current clicked row	Long	Protected	Internal
il_prevclickedrow	Tracks the number of the last clicked row	Long	Protected	Internal
il_prevrow	Previous row	Long	Protected	Internal
il_prevrowcount	Row count before last row focus change	Long	Protected	Internal
MULTIPLE	Constant set to 1	Integer	Public	Use with of_SetStyle
SINGLE	Constant set to 0	Integer	Public	Use with of_SetStyle

Events

N_cst_dwsrv_rowselection includes precoded events:

pfc_Clicked	pfc_RButtonDown
pfc_LButtonDown	pfc_RButtonUp
pfc_LButtonUp	pfc_RowFocusChanged

pfc_Clicked

Description

Selects or clears rows based on the pressed keys and the row selection option.

Syntax

dwcontrol.instancename.Event **pfc_Clicked** (*x*, *y*, *row*, *dwobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
x	Integer indicating the *x* position of the mouse click. From within the pfc_Clicked event, access this value through the *ai_xpos* argument
y	Integer indicating the *y* position of the mouse click. From within the pfc_Clicked event, access this value through the *ai_ypos* argument
row	Long indicating the clicked row. From within the pfc_Clicked event, access this value through the *al_row* argument
dwobject	DWObject variable indicating the clicked object (passed by reference). From within the pfc_Clicked event, access this value through the *adwo_obj* argument

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This event is called by the u_dw Clicked event.

pfc_LButtonDown

Description

Processes row selection when the user presses the left mouse button.

Syntax

dwcontrol.instancename.Event **pfc_LButtonDown** (*flags*, *x*, *y*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)

Argument	Description
flags	Unsigned long indicating the modifier keys and mouse buttons that are pressed. Values are: 1—Left mouse button 2—Right mouse button 4— SHIFT key 8— CTRL key 16—Middle mouse button From within the pfc_LButtonDown event, access this value through the *aul_flags* argument
x	Integer indicating the *x* position of the mouse click. From within the pfc_LButtonDown event, access this value through the *ai_xpos* argument
y	Integer indicating the *y* position of the mouse click. From within the pfc_LButtonDown event, access this value through the *ai_ypos* argument

Usage

This event is called by the u_dw LButtonDown event.

pfc_LButtonUp

Description

Processes row selection when the user releases the left mouse button.

Syntax

dwcontrol.instancename.Event **pfc_LButtonUp** (*flags, x, y*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
flags	Unsigned Long indicating the modifier keys and mouse buttons that are pressed. Values are: 1—Left mouse button 2—Right mouse button 4— SHIFT key 8— CTRL key 16—Middle mouse button From within the pfc_LButtonUp event, access this value through the *aul_flags* argument
x	Integer indicating the *x* position of the mouse click. From within the pfc_LButtonUp event, access this value through the *ai_xpos* argument

Argument	Description
y	Integer indicating the *y* position of the mouse click. From within the pfc_LButtonUp event, access this value through the *ai_ypos* argument

Usage

This event is called by the u_dw LButtonUp event.

pfc_RButtonDown

Description

Processes row selection when the user presses the right mouse button.

Syntax

dwcontrol.instancename.Event **pfc_RButtonDown** (*flags*, *x*, *y*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
flags	Unsigned Long indicating the modifier keys and mouse buttons that are pressed. Values are: 1—Left mouse button 2—Right mouse button 4— SHIFT key 8— CTRL key 16—Middle mouse button From within the pfc_RButtonDown event, access this value through the *aul_flags* argument
x	Integer indicating the *x* position of the mouse click. From within the pfc_RButtonDown event, access this value through the *ai_xpos* argument
y	Integer indicating the *y* position of the mouse click. From within the pfc_RButtonDown event, access this value through the *ai_ypos* argument

Return value

None

Usage

This event is called by the u_dw RButtonDown event.

pfc_RButtonUp

Description

Processes row selection when the user releases the right mouse button.

Syntax

dwcontrol.instancename.Event **pfc_RButtonUp** (*flags, x, y*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
flags	Unsigned Long indicating the modifier keys and mouse buttons that are pressed. Values are: 1—Left mouse button 2—Right mouse button 4— SHIFT key 8— CTRL key 16—Middle mouse button From within the pfc_RButtonUp event, access this value through the *aul_flags* argument
x	Integer indicating the *x* position of the mouse click. From within the pfc_RButtonUp event, access this value through the *ai_xpos* argument
y	Integer indicating the *y* position of the mouse click. From within the pfc_RButtonUp event, access this value through the *ai_ypos* argument

Return value

None

Usage

This event is called by the u_dw RButtonUp event.

pfc_RowFocusChanged

Description

Implements keyboard support for row selection.

Syntax

dwcontrol.instancename.Event **pfc_RowFocusChanged** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
row	Long specifying the new row. From within the pfc_RowFocusChanged event, access this value through the *al_row* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

This event is called by the u_dw RowFocusChanged event.

Functions

N_cst_dwsrv_rowselection contains precoded object functions:

of_ButtonUp	of_RowSelect
of_GetInfo	of_RowSelectExt
of_GetPropertyInfo	of_RowSelectMulti
of_GetStyle	of_RowSelectSingle
of_InvertSelection	of_SelectedCount
of_IsKeyboard	of_SetKeyboard
of_KeybRowSelect	of_SetStyle
of_KeybRowSelectExt	

of_ButtonUp

Description

Performs extended selection processing.

Access

Protected

Syntax

dwcontrol.instancename.**of_ButtonUp** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_RButtonUp event:

```
ib_buttonpressed = FALSE
of_ButtonUp( )
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

*dwcontrol.instancename.***of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_rowselect.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

*dwcontrol.instancename.***of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_rowselect.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetStyle

Description

Retrieves the currently enabled selection option.

Access

Public

Syntax

dwcontrol.instancename.**of_GetStyle** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)

Return value

Integer. Returns one of the following values:

- **SINGLE or 0** Single-row selection
- **MULTIPLE or 1** Multirow selection
- **EXTENDED or 2** Extended selection

Examples

This example calls the of_GetStyle function from within a DataWindow:

```
IF IsValid(inv_rowselect) THEN
   IF inv_rowselect.of_GetStyle( ) = 2 THEN
...
```

of_InvertSelection

Description

Reverses selection on all rows. Selected rows become unselected; unselected rows become selected.

Access

Public

Syntax

dwcontrol.instancename.**of_InvertSelection** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_InvertSelection function:

```
dw_emplist.inv_rowselect.of_InvertSelection()
```

of_IsKeyboard

Description

Reports whether keyboard support is enabled.

Access

Public

Syntax

dwcontrol.instancename.**of_IsKeyboard** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)

Return value

Boolean. Returns TRUE if keyboard support is enabled and FALSE if it is not.

Examples

This example calls the of_IsKeyboard function:

```
IF dw_1.inv_rowselect.of_IsKeyboard() THEN
   MessageBox("RowSelect", "Keyboard is enabled")
ELSE
   MessageBox("RowSelect", "Keyboard is disabled")
END IF
```

of_KeybRowSelect

Description

Performs the appropriate keyboard-based selection processing for the passed row.

Access

Protected

Syntax

dwcontrol.instancename.**of_KeybRowSelect** (*row*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)
row	Long specifying the row to process

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_RowFocusChanged event:

```
...
If (Not lb_rowdeletion) And (Not lb_rowclicked) Then
      li_rc = of_KeybRowSelect(al_row)
   End If
...
```

of_KeybRowSelectExt

Description

Performs keyboard-based extended selection.

Access

Protected

Syntax

dwcontrol.instancename.**of_KeybRowSelectExt** (*row, ctrlpressed, shiftpressed*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)
row	Long specifying the current row number
ctrlpressed	Boolean indicating whether the CTRL key is down

Argument	Description
shiftpressed	Boolean indicating whether the SHIFT key is down

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_KeybRowSelect function:

```
...
CHOOSE CASE ii_style
   CASE SINGLE
      li_rc = of_RowSelectSingle(al_row)
   CASE MULTIPLE
      // No Action.
   CASE EXTENDED
      li_rc = of_KeyBRowSelectExt(al_row,
         Keydown(KeyControl!), Keydown(KeyShift!))
END CHOOSE
...
```

of_RowSelect

Description

Calls one of the row selection functions based on the specified selection option:

- **Single-row selection** of_RowSelectSingle
- **Multirow selection** of_RowSelectMulti
- **Extended selection** of_RowSelectExt

Access

Protected

Syntax

*dwcontrol.instancename.***of_RowSelect** (*clickedrow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
clickedrow	Long containing the number of the clicked row

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_Clicked event:

```
IF Not IsValid( idw_requestor ) Then Return
IF IsNull(adwo_obj) THEN Return
IF IsNull(al_row) OR al_row <=0 THEN Return
of_RowSelect(al_row)
```

of_RowSelectExt

Description

Selects or clears rows depending on whether the CTRL and SHIFT keys are down.

Access

Protected

Syntax

dwcontrol.instancename.**of_RowSelectExt** (*clickedrow*, *ctrlpressed*, *shiftpressed*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
clickedrow	Long containing the number of the clicked row
ctrlpressed	Boolean indicating whether the CTRL key is down
shiftpressed	Boolean indicating whether the SHIFT key is down

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_RowSelect function:

```
...
CASE EXTENDED
   li_rc = of_RowSelectExt(al_row, &
      KeyDown(KeyControl!), Keydown(KeyShift!) )
...
```

of_RowSelectMulti

Description

Selects or clears the clicked row depending on its current state. This function does not clear previously selected rows.

Access

Protected

Syntax

dwcontrol.instancename.**of_RowSelectMulti** (*clickedrow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
clickedrow	Long containing the number of the clicked row

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_RowSelect function:

```
...
CASE MULTIPLE
   li_rc = of_RowSelectMulti(al_row)
...
```

of_RowSelectSingle

Description

Selects the clicked row, clearing previously selected rows.

Access

Protected

Syntax

dwcontrol.instancename.**of_RowSelectSingle** (*clickedrow*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
clickedrow	Long containing the number of the clicked row

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_RowSelect function:

```
Integer  li_rc
```

```
CHOOSE CASE ii_style
   CASE SINGLE
      li_rc = of_RowSelectSingle(al_row)
...
```

of_SelectedCount

Description

Retrieves the number of selected rows in a DataWindow.

Access

Public

Syntax

dwcontrol.instancename.**of_SelectedCount** (*selectedrows*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
selectedrows	Long array into which the function places the row numbers of selected rows (passed by reference)

Return value

Long. Returns the number of selected rows in the DataWindow if the function succeeds and -1 if an error occurs.

Examples

This example is from the pfc_Print event:

```
...
IF this.object.datawindow.print.page.range = &
   "selection" THEN
      lb_rowselection = IsValid(inv_rowselect)
      IF NOT lb_rowselection THEN
         of_SetRowSelect(TRUE)
      END IF
      ll_selectedcount = &
         inv_rowselect.of_SelectedCount(ll_selected)
      IF NOT lb_rowselection THEN
         of_SetRowSelect (FALSE)
      END IF
...
```

of_SetKeyboard

Description

Specifies whether keyboard selection is enabled.

Access

Public

Syntax

*dwcontrol.instancename.***of_SetKeyboard** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)
boolean	Boolean indicating whether to enable (TRUE) or disable (FALSE keyboard support

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

When keyboard selection is enabled, users can select multiple rows using the CTRL and SHIFT keys in conjunction with the arrow keys.

Examples

This example calls the of_SetKeyboard function:

```
dw_1.of_SetRowSelect(TRUE)
dw_1.inv_rowselect.of_SetKeyboard(FALSE)
```

of_SetStyle

Description

Specifies the row selection style:

- Single-row
- Multirow
- Extended

Access

Public

Syntax

*dwcontrol.instancename.***of_SetStyle** (*selectoption*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)

Argument	Description
selectoption	Integer or n_cst_dwsrv_rowselection constant specifying the row selection option: ♦ **SINGLE or 0** Single-row selection (default) ♦ **MULTIPLE or 1** Multirow selection ♦ **EXTENDED or 2** Extended row selection

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetStyle function:

```
dw_emplist.of_SetRowSelect(TRUE)
dw_emplist.inv_rowselect.of_SetStyle  &
   (dw_emplist.inv_rowselect.EXTENDED)
```

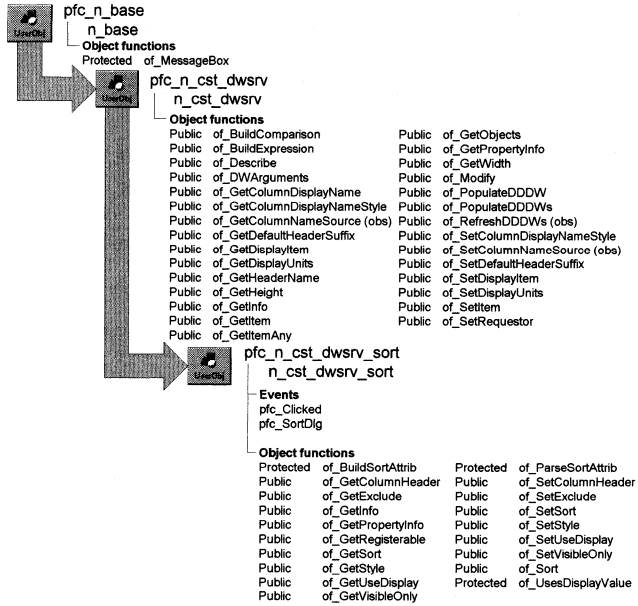
n_cst_dwsrv_sort

Description

Sort service. Provides functions that make it easy for you to add DataWindow sort functionality to an application. This service provides many options:

- You can choose among four styles of sort dialog boxes
- You can allow the user to sort by clicking on column headings

Ancestry



Library

PFCDWSRV.PBL
PFEDWSRV.PBL

Object relationships

n_cst_infoattrib
n_cst_propertyattrib
n_cst_returnattrib
n_cst_sortattrib
n_cst_string

Usage

Use this service to add sort capabilities to your application.

To use this service:

1 Enable the service using the u_dw of_SetSort function:

```
dw_emp.of_SetSort(TRUE)
```

2 (Optional) Specify whether to use display values instead of data values by calling the of_SetUseDisplay function:

```
dw_emp.inv_sort.of_SetUseDisplay(TRUE)
```

3 (Optional) Specify whether sort dialog boxes should display database column names, DataWindow column names, or column header names, by calling the n_cst_dwsrv of_SetColumnNameSource function:

```
dw_emp.inv_sort.of_SetColumnnameSource  &
   (dw_emp.inv_sort.HEADER)
```

4 (Optional) Specify the sort dialog box style by calling the of_SetStyle function:

```
dw_emp.inv_sort.of_SetStyle  &
   (dw_emp.inv_sort.DRAGDROP)
```

5 (Optional) Enable column header sorting by calling the of_SetColumnHeader function:

```
dw_emp.inv_sort.of_SetColumnHeader(TRUE)
```

6 Display the sort dialog box by calling the u_dw pfc_SortDlg event:

```
dw_emp.Event pfc_SortDlg()
```

See also

n_cst_dwsrv_filter
u_dw

Instance variables

N_cst_dwsrv_sort includes instance variables:

Instance variable	Description	Data type	Access	Usage
DRAGDROP	Constant set to 1	Integer	Public	Use with of_SetStyle
DROPDOWNLISTBOX	Constant set to 3	Integer	Public	Use with of_SetStyle
ib_columnheadersort	Tracks whether column header sort is active	Boolean	Protected	Internal
ib_displayvalues	Tracks whether display values are used for sorting	Boolean	Protected	Internal

Instance variable	Description	Data type	Access	Usage
ib_visibleonly	Tracks whether sort dialog boxes display visible columns only	Boolean	Protected	Internal
ii_style	Tracks sort style	Integer	Protected	Internal
is_excludecolumns[ ]	Contains a list of columns not displayed in sort dialog boxes	String	Protected	Internal
is_sortcolumn	Indicates the sort column when sorting by clicking headings	String	Protected	Internal
is_sortorder	Indicates sort order when the user sorts by clicking headings	String	Protected	Internal
SIMPLE	Constant set to 2	Integer	Public	Use with of_SetStyle

Events

N_cst_dwsrv_sort includes user events:

pfc_Clicked
pfc_SortDlg

pfc_Clicked

Description

Sorts rows when the user clicks on the header. This event executes only if you have enabled column header sorting by calling the of_SetColumnHeader function.

Syntax

dwcontrol.instancename.Event **pfc_Clicked** (*x, y, row, dwobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
x	Integer indicating the *x* position of the mouse click. From within the pfc_Clicked event, access this value through the *ai_xpos* argument

Argument	Description
y	Integer indicating the *y* position of the mouse click. From within the pfc_Clicked event, access this value through the *ai_ypos* argument
row	Long indicating the clicked row. From within the pfc_Clicked event, access this value through the *al_row* argument
dwobject	DWObject variable indicating the clicked object (passed by reference). From within the pfc_Clicked event, access this value through the *adwo_obj* argument

Return value

Integer. Returns 1 if the event succeeds, 0 if no action is required, and -1 if an error occurs.

Usage

This event is called by the u_dw clicked event.

pfc_SortDlg

Description

Displays a sort dialog box and sorts the DataWindow as specified by the user.

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

Call this event to display a sort dialog box using the style specified in the of_SetStyle function.

Functions

N_cst_dwsrv_sort includes precoded object functions:

of_BuildSortAttrib	of_ParseSortAttrib
of_GetColumnHeader	of_SetColumnHeader
of_GetExclude	of_SetExclude
of_GetInfo	of_SetSort
of_GetPropertyInfo	of_SetStyle
of_GetRegisterable	of_SetUseDisplay
of_GetSort	of_SetVisibleOnly
of_GetStyle	of_Sort
of_GetUseDisplay	of_UsesDisplayValue
of_GetVisibleOnly	

of_BuildSortAttrib

Description

Prepares sort information before displaying one of the sort dialog boxes.

Access

Protected

Syntax

dwcontrol.instancename.**of_BuildSortAttrib** (*sortattrib*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
sortattrib	N_cst_sortattrib instance into which the function places sort information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SetSort function:

```
...
CASE DRAGDROP, SIMPLE, DROPDOWNLISTBOX
   IF of_BuildSortAttrib(lnv_sortattrib) <> 1 THEN &
      Return -1
   END IF
...
```

of_GetColumnHeader

Description

Reports whether the sort service is set to sort when the user clicks on a column header.

Access

Public

Syntax

dwcontrol.instancename.**of_GetColumnHeader** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Return value

Boolean. Returns TRUE if column header sort is active and FALSE if it is not.

Examples

This example is from the pfc_Clicked event:

```
...
IF NOT of_GetColumnHeader() THEN Return 0
...
```

of_GetExclude

Description

Retrieves the list of excluded columns, as set by the of_SetExclude function.

Access

Public

Syntax

dwcontrol.instancename.**of_GetExclude** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
columns	String array into which the function places the excluded columns (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetExclude function:

```
String    ls_exclude[ ]
String    ls_list
Integer   li_count

dw_emp.inv_sort.of_GetExclude(ls_exclude)
For li_count = 1 to UpperBound(ls_exclude)
   ls_list += ls_exclude[li_count] + ", "
NEXT
MessageBox("Sort Service", &
   "Excluded columns are: " + ls_list)
```

of_GetInfo

Description

Retrieves service information.

Access

Public

Syntax

dwcontrol.instancename.**of_GetInfo** (*infoobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
infoobject	N_cst_infoattrib instance into which the function places service information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetInfo function:

```
n_cst_infoattrib  lnv_info

dw_1.inv_sort.of_GetInfo(lnv_info)
MessageBox("Info",  &
  "Description: " + lnv_info.is_description  &
    + ". Name: " + lnv_info.is_name)
```

of_GetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.**of_GetPropertyInfo** (*propertyobject*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
propertyobject	N_cst_propertyattrib instance into which the function places property information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The DataWindow Properties window calls this function to access service information.

Examples

This example calls the of_GetPropertyInfo function:

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_sort.of_GetPropertyInfo(lnv_prop)
MessageBox("Info",  &
  "Description: " + lnv_prop.is_description  &
    + ". Name: " + lnv_prop.is_name  &
      + ". Property tab text: " +  &
        lnv_prop.is_propertytabtext)
```

of_GetRegisterable

Description

Retrieves an array of registerable columns.

Access

Public

Syntax

dwcontrol.instancename.**of_GetRegisterable** (*columns*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
columns	String array into which the function places a list of registerable columns (passed by reference)

Return value

Integer. Returns the number of entries in the *columns* array.

Examples

This example calls the of_GetRegisterable function:

```
String   ls_columns[ ]
Integer  li_count, li_loop
li_loop =  &
   dw_1.inv_sort.of_GetRegisterable  &
     (ls_columns)
lb_1.Reset()
IF li_loop <> 0 THEN
  FOR li_count = 1 to li_loop
    lb_1.AddItem(ls_columns[li_count])
  NEXT
ELSE
  lb_1.AddItem("No Registerable columns")
END IF
```

of_GetSort

Description

Retrieves the current sort expression for the DataWindow. The sort is created when you call the of_SetSort function.

Access

Public

Syntax

dwcontrol.instancename.**of_GetSort** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Return value

String. Returns the current sort expression for the DataWindow.

Examples

This example calls the of_GetSort function:

```
String    ls_sort

ls_sort = &
   dw_emp.inv_sort.of_GetSort()
MessageBox("Sort Service", "Sort is: " &
   + ls_sort)
```

of_GetStyle

Description

Retrieves the current sort style (the sort style specifies which dialog box displays when you call the of_Sort function).

Access

Public

Syntax

dwcontrol.instancename.**of_GetStyle** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Return value

Integer. Returns the current sort style:

- **DEFAULT or 0** PowerBuilder Sort dialog box (default)
- **DRAGDROP or 1** PFC's w_sortdragdrop dialog box

- **SIMPLE or 2** PFC's w_sortsingle dialog box
- **DROPDOWNLISTBOX or 3** PFC's w_sortmulti dialog box

Examples

This example calls the of_GetStyle function:

```
IF IsValid(inv_sort) THEN
   IF inv_sort.of_GetStyle( )= 2 THEN
...
```

of_GetUseDisplay

Description

Reports whether this service is using display values for PFC sort dialog boxes.

PowerBuilder Sort dialog box always uses data values
The PowerBuilder Sort dialog box always uses data values for sorting.

Access

Public

Syntax

dwcontrol.instancename.**of_GetUseDisplay** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Return value

Boolean. Returns TRUE if PFC dialog boxes use display values and FALSE if they use data values.

Examples

This example calls the of_GetUseDisplay function:

```
...
IF idw_dw.inv_sort.of_GetUseDisplay( ) AND &
   idw_dw.inv_sort.of_UsesDisplayValue &
      (ls_col) THEN
         ls_sort_string = ls_sort_string + &
            "LookUpDisplay(" + ls_col + ") "
ELSE
...
```

of_GetVisibleOnly

Description

Reports whether the sort service uses visible columns only.

Access

Public

Syntax

dwcontrol.instancename.**of_GetVisibleOnly** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Return value

Boolean. Returns TRUE if the sort service uses visible columns only and FALSE if it uses all columns.

Examples

This example calls the of_GetVisibleOnly function:

```
IF dw_emp.inv_sort.of_GetVisibleOnly() THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      ("Visible columns only")
ELSE
   gnv_app.of_GetFrame().SetMicroHelp &
      ("All columns used for sort")
END IF
```

of_ParseSortAttrib

Description

Parses a sort statement into its component parts.

Access

Protected

Syntax

dwcontrol.instancename.**of_ParseSortAttrib** (*originalsort*, *sortattrib*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
originalsort	String containing the original sort statement
sortattrib	N_cst_sortattrib instance into which the function places sort information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_BuildSortAttrib function:

```
...
of_ParseSortAttrib(anv_sortattrib.is_sort, &
    anv_sortattrib)
...
```

of_SetColumnHeader

Description

Specifies whether n_cst_dwsrv_sort sorts when the user clicks on a column header. With column header sorting (a feature found in many Windows 95-compliant applications), PFC sorts a DataWindow based on the column the user clicks on. If the user clicks the same column twice, PFC reverses the sort order.

Access

Public

Syntax

dwcontrol.instancename.**of_SetColumnHeader** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
boolean	Boolean specifying whether the service implements column header sorting (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetColumnHeader function:

```
dw_list.inv_sort.of_SetColumnHeader(TRUE)
```

of_SetExclude

Description

Establishes a list of columns to be excluded from sort dialog boxes.

Access

Public

Syntax

dwcontrol.instancename.**of_SetExclude** (*excludecols*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
excludecols	String array specifying columns to be excluded from sort dialog boxes

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetExclude function:

```
String    ls_exclude[ ]

ls_exclude[1] = "dept_id"
dw_emp.inv_sort.of_SetExclude(ls_exclude)
```

of_SetSort

Description

Specifies a DataWindow sort string or opens a sort dialog box.

Access

Public

Syntax

dwcontrol.instancename.**of_SetSort** (*sort*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
sort	String containing a valid DataWindow sort expression. Column names must be preceded by a pound sign (#). If this argument is NULL, the function displays a sort dialog box

Return value

Integer. Returns 1 if the function succeeds, 0 if the user canceled out of the sort dialog box, and -1 if an error occurs.

Usage

To control the type of sort dialog box, call the of_SetStyle function.

Examples

This example calls the of_SetSort function:

```
String    ls_null
Integer   li_return
```

```
SetNull(ls_null)
dw_emp.inv_sort.of_SetStyle(1)
li_return = &
   dw_emp.inv_sort.of_SetSort(ls_null)
gnv_app.of_GetFrame().SetMicroHelp &
   ("Return code = " + String(li_return))
dw_emp.inv_sort.of_Sort( )
```

of_SetStyle

Description

Specifies the sort style. The sort style determines the dialog box that displays when the of_SetSort function is called:

Default PowerBuilder Sort dialog box
W_sortdragdrop dialog box
W_sortsingle dialog box
W_sortmulti dialog box

Access

Public

Syntax

dwcontrol.instancename.**of_SetStyle** (*sortstyle*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
sortstyle	Integer or n_cst_dwsrv_sort constant specifying the sort style: ◆ **DEFAULT or 0** PowerBuilder Sort dialog box (default) ◆ **DRAGDROP or 1** W_sortdragdrop dialog box ◆ **SIMPLE or 2** W_sortsingle dialog box ◆ **DROPDOWNLISTBOX or 3** W_sortmulti dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to specify a sort style before calling the of_SetSort function.

To implement column header sorting (a feature found in many Windows 95-compliant applications), call the of_SetColumnHeader function.

Examples

This example calls the of_SetStyle function to enable display of the w_sortdragdrop dialog box (of_SetColumnNameSource is an ancestor function, defined in n_cst_dwsrv):

```
dw_emplist.of_SetSort(TRUE)
dw_emplist.inv_sort.of_SetColumnNameSource(2)
dw_emplist.inv_sort.of_SetUseDisplay(TRUE)
dw_emplist.inv_sort.of_SetStyle  &
  (dw_emplist.inv_sort.DRAGDROP)
```

of_SetUseDisplay

Description

Specifies whether PFC sorts on display values or data values, for columns that display using a code table.

PowerBuilder Sort dialog box always uses data values
The PowerBuilder Sort dialog box always uses data values for sorting.

Access

Public

Syntax

dwcontrol.instancename.**of_SetUseDisplay** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
boolean	Boolean specifying whether to use display values (TRUE) or data values (FALSE) for sorting

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetUseDisplay function to specify that PFC sort dialog boxes uses display values for sorting (of_SetColumnNameSource is an ancestor function, defined in n_cst_dwsrv):

```
dw_emplist.of_SetSort(TRUE)
dw_emplist.of_SetStyle(dw_emplist.inv_sort.SIMPLE)
dw_emplist.inv_sort.of_SetUseDisplay(TRUE)
```

of_SetVisibleOnly

Description

Controls whether sort dialog boxes display all columns or visible columns only.

Access

Public

Syntax

dwcontrol.instancename.**of_SetVisibleOnly** (*boolean*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
boolean	Boolean indicating whether sort dialog boxes display visible columns only (TRUE) or all columns (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetVisibleOnly function:

```
dw_emp.inv_sort.of_SetVisibleOnly(TRUE)
```

of_Sort

Description

Sorts a DataWindow and calls the u_dw pfc_RowChanged event.

Access

Public

Syntax

dwcontrol.instancename.**of_Sort** ( )

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to perform a sort automatically. Call the pfc_SortDlg event to display a sort dialog box.

Examples

This example is from the pfc_Clicked event:

```
...
li_rc = of_SetSort(ls_sortstring)
If li_rc < 0 Then Return li_rc

li_rc = of_Sort()
...
```

of_UsesDisplayValue

Description

Reports whether a column has a display value.

Access

Protected

Syntax

dwcontrol.instancename.**of_UsesDisplayValue** (*column*)

Argument	Description
dwcontrol	Instance name of the u_dw-based DataWindow control
instancename	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
column	String specifying the column name

Return value

Boolean. Returns TRUE if the column has a display value and FALSE if it does not.

Usage

Internal.

Examples

This example calls the of_UsesDisplayValue function:

```
IF idw_dw.inv_sort.of_GetUseDisplay() AND &
   idw_dw.inv_sort.of_UsesDisplayValue  &
      (ls_col) THEN
         ls_sort_string = ls_sort_string + &
            "LookUpDisplay(" + ls_col + ") "
...
```

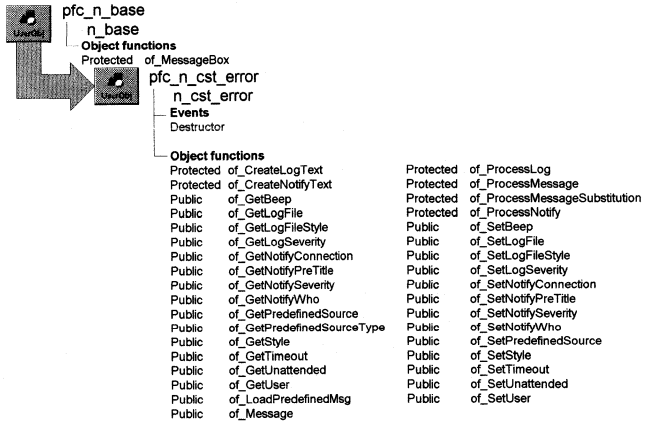
n_cst_error

Description

Error message service. Provides many features for handling your application's error messages, including:

- Functions to enable error message display for application, SQL, DataWindow, and system errors
- Message logging to a file or database
- Automatic error notification via e-mail (MAPI-compliant e-mail systems only)
- Displaying messages in w_message (the PFC message dialog box)
- Access to an error message database (which can reside in either a database or a file)
- Symbolic parameter replacement

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

d_definedmessages
Messages table in PFC.DB (copy to your application database)
n_cst_conversion
n_cst_errorattrib
n_cst_filesrv
n_cst_string
n_ds
n_tr
w_message

Usage

To use n_cst_error:

1 Create the object by calling the n_cst_appmanager of_SetError function (this example is from an n_cst_appmanager pfc_Open event:

```
this.of_SetError(TRUE)
```

2 Specify the error message source:

- If the source is a file, call the of_SetPredefinedSource function, passing the filename:

```
this.inv_error.of_SetPredefinedSource &
   ("c:\eisapp\eiserr.txt")
```

- If the source is a database, call the of_SetPredefinedSource function, passing the transaction object:

```
this.inv_error.of_SetPredefinedSource &
   (itr_error)
```

3 Specify the name of the log file:

```
this.inv_error.of_SetLogFile &
   ("c:\pb6\errlog.txt")
```

4 Specify the types of messages for which the error service will provide automatic notification and logging:

```
this.inv_error.of_SetNotifySeverity(5)
this.inv_error.of_SetLogSeverity(4)
```

5 If your application uses the error service's automatic notification feature, specify a MailSession variable to use. Also specify the e-mail IDs of the users to be notified automatically. This example assumes a mechanism for storing e-mail IDs and user passwords:

```
this.inv_error.of_SetNotifyConnection &
   (ims_mailsess)
this.inv_error.of_SetNotifyWho(is_autonotify  &
   is_notifylist)
```

N_cst_appmanager pfc_Open
The steps listed above might all be coded in the n_cst_appmanager pfc_Open event.

6 In your application error checking, call the of_Message function to display messages, with optional logging and notification. The of_Message function allows you to either use the message database or specify message text dynamically. This example uses the message database:

```
gnv_app.inv_error.of_Message &
   ("EIS0210")
```

See also n_cst_errorattrib

Instance variables

N_cst_error includes instance variables:

Instance variable	Description	Data type	Access	Usage
DATABASE	Constant set to "database"	String	Protected	Internal
DEFAULT	Constant set to 0	Integer	Public	Use with of_SetStyle
FILE	Constant set to "File"	String	Protected	Internal
ib_beep	Indicates whether PFC beeps when displaying a message box	Boolean	Protected	Set with of_SetBeep
ib_unattended	Indicates whether the unattended option is enabled	Boolean	Protected	Set with of_SetUnattended
ids_messages	DataStore that caches all messages at runtime	n_ds	Protected	Internal
ii_logfilestyle	Controls format of log file text	Integer	Protected	Set with of_SetLogFileStyle
ii_logseverity	Error logging threshold	Integer	Protected	Set with of_SetLogSeverity

Instance variable	Description	Data type	Access	Usage
ii_notifyseverity	Automatic notification threshold	Integer	Protected	Set with of_SetNotifySeverity
ii_style	Controls whether the service uses the PowerBuilder MessageBox (default) or the PFC w_message window	Integer	Protected	Set with of_SetStyle
ii_timeout	Timeout interval	Integer	Protected	Set with of_SetTimeout
ims_mses	MailSession for automatic error notification	MailSession	Protected	Set with of_SetNotifyConnection
inv_conversion	Reference variable for the conversion service	n_cst_conversion	Protected	Internal
inv_errorpass	Used to pass information to error message boxes	n_cst_errorattrib	Protected	Internal
inv_filesrv	Reference variable for the file service	n_cst_filesrv	Protected	Internal
inv_string	Reference variable for the string service	n_cst_string	Protected	Internal
is_logfile	File to which the service writes messages	String	Protected	Set with of_SetLogFile
is_msgfile	File containing predefined error messages	String	Protected	Set with of_SetPredefinedSource (passing a string)
is_msgsrc	Input source for predefined messages: file or database	String	Protected	Set with of_SetPredefinedSource
is_notifyaddress[ ]	List of e-mail addresses to be notified when errors occur	String	Protected	Set with of_SetNotifyWho
is_notifypretitle	Prefix used with mail notification	String	Protected	Set with of_SetNotifyPreTitle

Instance variable	Description	Data type	Access	Usage
is_notifywho[ ]	List of e-mail IDs to be notified when errors occur	String	Protected	Set with of_SetNotifyWho
is_user	Prefix ;used with mail notification	String	Protected	Set with of_Message or through error message database
itr_msgdb	Transaction object for message database	n_tr	Public	Set with of_SetPredefinedSource (passing a Transaction object)
NEWLINE_DELIMITED	Constant set to 2	Integer	Public	Use with of_SetLogFileStyle
PFCWINDOW	Constant set to 1	Integer	Public	Use with of_SetStyle
TAB_DELIMITED	Constant set to 1	Integer	Public	Use with of_SetLogFileStyle

Events

N_cst_error includes one precoded event:

Destructor

Destructor

Description — Destroys the message DataStore and the file service object.

Usage — This event executes when the object is destroyed.

Functions

N_cst_error includes precoded object functions:

of_CreateLogText	of_ProcessLog
of_CreateNotifyText	of_ProcessMessage
of_GetBeep	of_ProcessMessageSubstitution

of_GetLogFile	of_ProcessNotify
of_GetLogFileStyle	of_SetBeep
of_GetLogSeverity	of_SetLogFile
of_GetNotifyConnection	of_SetLogFileStyle
of_GetNotifyPreTitle	of_SetLogSeverity
of_GetNotifySeverity	of_SetNotifyConnection
of_GetNotifyWho	of_SetNotifyPreTitle
of_GetPredefinedSource	of_SetNotifySeverity
of_GetPredefinedSourceType	of_SetNotifyWho
of_GetStyle	of_SetPredefinedSource
of_GetTimeOut	of_SetStyle
of_GetUnattended	of_SetTimeOut
of_GetUser	of_SetUnattended
of_LoadPredefinedMsg	of_SetUser
of_Message	

of_CreateLogText

Description

Creates text used to perform the logging process.

Access

Protected

Syntax

instancename.**of_CreateLogText** (*text*)

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
text	String containing text to be logged

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

To log anything other than standard text, override this function in a descendant.

Examples

This example is from the of_ProcessLog function:

```
...
li_rc = of_CreateLogText(ls_errortext)
...
```

of_CreateNotifyText

Description

Creates text used by the notify process.

Access

Protected

Syntax

instancename.**of_CreateNotifyText** (*subject, text*)

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
subject	String containing text used in the mail messages subject line
text	String containing message text

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

To notify with anything other than standard text, override this function in a descendant.

Examples

This example is from the of_ProcessNotify function:

```
...
li_rc = of_CreateNotifyText &
   (ls_notesubject, ls_notetext)
...
```

of_GetBeep

Description

Reports whether PFC beeps before displaying the w_message window.

Access

Public

Syntax

application.instancename.**of_GetBeep** ( )

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Boolean. Returns TRUE if PFC beeps before displaying the w_message window and FALSE if it does not.

Examples

This example calls the of_GetBeep function:

```
IF gnv_app.inv_error.of_GetBeep() THEN
   MessageBox("Error", "Beep enabled"
ELSE
   MessageBox("Error", "Beep disabled"
END IF
```

of_GetLogFile

Description

Retrieves the name of the log file.

Access

Public

Syntax

application.instancename.**of_GetLogFile**( )

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

String. Returns the name of the log file.

Examples

This example calls the of_GetLogFile function:

```
String    ls_logfile

ls_logfile = gnv_app.inv_error.of_GetLogFile()
...
```

of_GetLogFileStyle

Description

Retrieves the log file style, which specifies the format of the log file.

Access

Public

Syntax

application.instancename.**of_GetLogFileStyle** ( )

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)

Return value

Integer. Returns the log file style:

- **TAB_DELIMITED or 1** Tab delimited log file
- **NEWLINE_DELIMITED or 2** Newline delimited log file

Examples

This example calls the of_GetLogFileStyle function:

```
Integer li_style

li_style = gnv_app.inv_error.of_GetLogFileStyle()
IF li_style = gnv_app.inv_error.TAB_DELIMITED THEN
   MessageBox("Log File Style", "Tab delimited")
ELSE
   MessageBox("Log File Style", "Newline delimited")
END IF
```

of_GetLogSeverity

Description

Retrieves the current log severity level.

Access

Public

Syntax

application.instancename.**of_GetLogSeverity ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns the log severity level.

Examples

This example calls the of_GetLogSeverity function:

```
Integer   ii_severity

ii_severity = gnv_app.inv_error.of_GetLogSeverity( )
...
```

of_GetNotifyConnection

Description

Retrieves the mail session used for automatic notification.

Access

Public

Syntax

application.instancename.**of_GetNotifyConnection** (*mailsession*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
mailsession	MailSession variable into which the function places the current mail connection (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetNotifyConnection function:

```
MailSession   lms_mses
Integer       li_return

li_return = &
   gnv_app.inv_error.of_GetNotifyConnection &
      (lms_mses)
...
```

of_GetNotifyPreTitle

Description

Retrieves the prefix used with automatic e-mail notification.

Access

Public

Syntax

application.instancename.**of_GetNotifyPreTitle** ( )

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)

Return value

String. Returns the prefix used with automatic e-mail notification.

Examples

This example calls the of_GetNotifyPreTitle function:

```
String  ls_prefix
```

```
ls_prefix = gnv_app.inv_error.of_GetNotifyPreTitle()
MessageBox("Prefix", "Prefix is " + ls_prefix)
```

of_GetNotifySeverity

Description: Retrieves the current notify severity level.

Access: Public

Syntax: *application.instancename*.**of_GetNotifySeverity** ( )

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value: Integer. Returns the current notify severity.

Examples: This example calls the of_GetNotifySeverity function:

```
Integer   ii_severity

ii_severity = &
   gnv_app.inv_error.of_GetNotifySeverity()
...
```

of_GetNotifyWho

Description: Retrieves a list of users to be notified when errors occur.

Access: Public

Syntax: *application.instancename*.**of_GetNotifyWho** (*users*, *addresses*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Argument	Description
users	String array into which the function places the list of users to be notified when errors occur (passed by reference)
addresses	String array into which the function places the list of e-mail addresses to be notified when errors occur (passed by reference)

Return value

Integer. Returns the number of users and addresses to notify.

Examples

This example calls the of_GetNotifyWho function:

```
String    ls_notify[ ]
Integer   li_count, li_return

li_return = &
   gnv_app.inv_error.of_GetNotifyWho(ls_notify)

FOR li_count = 1 to li_return
   lb_who.Item[li_count] = ls_notify[li_count]
NEXT
```

of_GetPredefinedSource

Description

Retrieves the source for predefined messages. This is either a file or a Transaction object connected to a database containing the messages.

Access

Public

Syntax

application.instancename.**of_GetPredefinedSource** (*source*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
source	String or n_tr variable for predefined messages

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetPredefinedSource function:

```
String    ls_source
Integer   li_return
```

```
li_return = &
   gnv_app.inv_error.of_GetPredefinedSource  &
      (ls_source)
...
```

of_GetPredefinedSourceType

Description

Reports the message source type: file or database.

Access

Public

Syntax

application.instancename.**of_GetPredefinedSourceType ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

String. Returns the message source type:

File
Database

Examples

This example calls the of_GetPredefinedSourceType function:

```
String   ls_source

ls_source = &
   gnv_app.inv_error.of_GetPredefinedSourceType()
...
```

of_GetStyle

Description

Retrieves the current message box style:

PowerBuilder MessageBox
PFC w_message window

Access

Public

Syntax

application.instancename.**of_GetStyle ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns the message box style:

- **DEFAULT or 0** PowerBuilder MessageBox
- **PFCWINDOW or 1** PFC w_message window

Examples

This example calls the of_GetStyle function:

```
Integer   li_return

li_return = gnv_app.inv_error.of_GetStyle()
...
```

of_GetTimeOut

Description

Retrieves the number of seconds before a w_message dialog box closes automatically.

Access

Public

Syntax

application.instancename.**of_GetTimeOut ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns the number of seconds before PFC automatically closes a w_message dialog box. A value of 0 indicates that PFC does not close w_message automatically.

Examples

This example calls the of_GetTimeOut function:

```
Integer   li_return
```

```
li_return = &
   gnv_app.inv_error.of_GetTimeOut( )
...
```

of_GetUnattended

Description: Reports whether the unattended option is enabled.

Access: Public

Syntax: *application.instancename*.**of_GetUnattended ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value: Boolean. Returns TRUE if the unattended option is enabled and FALSE if it is not.

Examples: This example calls the of_GetUnattended function:

```
...
IF gnv_app.inv_error.of_GetUnattended( ) THEN
   MessageBox("Error", "Unattended enabled")
ELSE
   MessageBox("Error", "Unattended disabled")
END IF
```

of_GetUser

Description: Retrieves the user name.

Access: Public

Syntax: *application.instancename*.**of_GetUser ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)

Return value

String. Returns the user name as specified by the of_SetUser function.

Usage

The user name is used in message logging and automatic e-mail notification.

Examples

This example calls the of_GetUser function:

```
String  ls_user

ls_user = gnv_app.inv_error.of_GetUser()
MessageBox("User", "User is " + ls_user)
```

of_LoadPredefinedMsg

Description

Loads or reloads the message cache from the source file or database.

Access

Public

Syntax

application.instancename.**of_LoadPredefinedMsg ( )**

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_LoadPredefinedMsg function:

```
Integer   li_return

li_return = &
   gnv_app.inv_error.of_LoadPredefinedMsg()
...
```

of_Message

Displays the w_message dialog box using either the ids_messages DataStore or a passed string. There are two syntaxes.

To	Use
Display a specified message from the ids_messages DataStore	Syntax 1
Display a message using a passed string	Syntax 2

Syntax 1 **Display message text from the ids_messages DataStore**

Description

Displays message text and display information from the ids_messages DataStore.

Access

Public

Syntax

application.instancename.**of_Message** (*messageid* {, *substitutions* {, *overridetitle* } })

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
messageid	String specifying the identifier of the message to be displayed. This value must be in the msgnumber column of the ids_messages DataStore
substitutions (optional)	String array specifying substitution parameters for the message. The message source indicates substitution parameters with %s
overridetitle (optional)	String specifying a title to override the default title

Return value

Integer. Returns the button the user clicked if the function succeeds and -1 if an error occurs.

Usage

Call this function to display a predefined message from the ids_messages DataStore.

This function displays either the PowerBuilder MessageBox or the PFC w_message window, depending on what you specify for the of_SetStyle function.

The function uses substitution parameters to replace percent sign characters (%) in the predefined message. You define one element in the *substitution* array for each % in the predefined message text; the first element in the array replaces the first %, the second element replaces the second %, and so on.

Use Syntax 2 to display w_message passing message text in a string.

Examples

This example calls the of_Message function:

```
gnv_app.inv_error.of_Message("EIS0210")
```

Syntax 2 **Display a message using a passed string**

Description

Displays message text and display information from passed arguments.

Access

Public

Syntax

application.instancename.**of_Message** (*title*, *message* {, *icon* {, *button* {, *default* {, *severity*, *print*, *userinput* } } } })

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
title	String specifying the title of the w_message dialog box
message	String specifying the message text
icon (optional)	Icon enumerated data type indicating the icon you want to display on the left side of the message box. Values are: ♦ Information! (Default) ♦ StopSign! ♦ Exclamation! ♦ Question! ♦ None!
button (optional)	Button enumerated data type indicating the set of CommandButtons you want to display at the bottom of the message box. The buttons are numbered in the order listed in the enumerated data type. Values are: ♦ OK! (default) ♦ OKCancel! ♦ YesNo! ♦ YesNoCancel! ♦ RetryCancel! ♦ AbortRetryIgnore!
default (optional)	Integer specifying the number of the button you want to be the default button. The default is 1. If you specify a number larger than the number of buttons displayed, w_message uses the default
severity (optional)	Integer indicating message severity

Argument	Description
print (optional)	Boolean indicating whether to display a print button
userinput (optional)	Boolean indicating whether the cb_userinput button displays, allowing the users to specify additional information

Return value

Integer. Returns the button the user clicked if the function succeeds and -1 if an error occurs.

Usage

Call this function to display a string in the w_message dialog box, optionally specifying additional information to be displayed.

This function displays either the PowerBuilder MessageBox or the PFC w_message window, depending on what you specify for the of_SetStyle function. The *print* and *userinput* arguments apply only when using the PFC w_message window.

Use Syntax 1 to display w_message accessing a message in the ids_messages DataStore.

Examples

This example calls the of_Message function:

```
Integer   li_return

li_return = &
   gnv_app.inv_error.of_Message &
      ("EIS Application", &
         "Data has changed. Call Sys Admin.", &
            Information!, Ok!, 1, 100, FALSE, FALSE)
```

of_ProcessLog

Description

Writes a message to the log file.

Access

Protected

Syntax

instancename.**of_ProcessLog** ( )

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_ProcessMessage function:

```
...
IF (inv_errorpass.ii_severity >= ii_logseverity) THEN
   li_testrc = of_ProcessLog( )
END IF
...
```

of_ProcessMessage

Description

Processes a message, logging it and notifying administrators as appropriate.

Access

Protected

Syntax

instancename.**of_ProcessMessage** ( )

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns the button pressed on the message window if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Message function:

```
...
Return of_ProcessMessage( )
```

of_ProcessMessageSubstitution

Description

Replaces substitution parameters in a message with the passed values.

Access

Protected

Syntax

instancename.**of_ProcessMessageSubstitution** (*parms*)

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Argument	Description
parms	String array containing the values to be substituted into the message replacing %s

Return value

Integer. Returns 1 if the function succeeds, 0 if it could not correctly convert the substitution, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Message function:

```
...
IF UpperBound (as_msgparms) > 0 THEN
    li_rc = &
        of_ProcessMessageSubstitution(as_msgparms)
END IF
...
```

of_ProcessNotify

Description

Sends an e-mail notification to the list of users specified in the of_SetNotifyWho function.

Access

Protected

Syntax

instancename.**of_ProcessNotify ( )**

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Return value

Integer. Returns 1 if the function succeeds, 0 if there is no mail connection or the list is empty, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_ProcessMessage function:

```
...
IF (inv_errorpass.ii_severity >= &
    ii_notifyseverity) THEN
        li_testrc = of_ProcessNotify( )
END IF
```

```
Return inv_errorpass.ii_buttonclicked
```

of_SetBeep

Description

Specifies whether PFC beeps before displaying the w_message window.

Access

Public

Syntax

*application.instancename.***of_SetBeep** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
boolean	Boolean specifying whether to beep (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBeep function:

```
gnv_app.inv_error.of_SetBeep(TRUE)
```

of_SetLogFile

Description

Specifies the name of the log file.

Access

Public

Syntax

*application.instancename.***of_SetLogFile** (*logfile*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
logfile	String specifying the name of the file to contain logged messages

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The function will create *logfile* if it doesn't exist. To disable logging, call this function passing an empty string. By default, PFC writes all loggable messages to errlog.txt.

N_cst_error appends messages to the end of the log file. You might consider purging it regularly.

Examples

This example calls the of_SetLogFile function:

```
gnv_app.inv_error.of_SetLogFile &
    ("c:\eisapp\eislog.txt")
```

of_SetLogFileStyle

Description

Specifies the formatting used for message logging.

Access

Public

Syntax

application.instancename.**of_SetLogFileStyle** (*logstyle*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)
logstyle	Integer or n_cst_error constant specifying the log file style: ♦ **TAB_DELIMITED or 1** Log entry items are separated by a tab ♦ **NEWLINE_DELIMITED or 2** Log entry items are separated by a newline character

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

A newline delimited log file is easier to read but contains more lines.

Examples

This example calls the of_SetLogFileStyle function:

```
...
this.inv_error.of_SetLogFileStyle  &
 (this.inv_error.NEWLINE_DELIMITED)
...
```

of_SetLogSeverity

Description

Specifies the log severity level.

Access

Public

Syntax

application.instancename.**of_SetLogSeverity** (*level*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
level	Integer specifying the logging threshold

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Messages have a default severity of zero. To log all messages, call this function passing -1. N_cst_error automatically logs messages with a severity higher than *level*.

Examples

This example calls the of_SetLogSeverity function:

```
gnv_app.inv_error.of_SetLogSeverity(5)
```

of_SetNotifyConnection

Description

Sets the mail session used for automatic notification.

Access

Public

Syntax

application.instancename.**of_SetNotifyConnection** (*mailsession*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
mailsession	MailSession variable containing a valid mail connection

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Create a valid mail connection by instantiating *mailsession* and calling the PowerScript MailLogon function.

32-bit PowerBuilder must use a 32-bit e-mail system; 16-bit PowerBuilder must use a 16-bit e-mail system.

Examples

This example calls the of_SetNotifyConnection function:

```
...
gnv_app.inv_error.of_SetNotifyConnection(ims_sess)
...
```

of_SetNotifyPreTitle

Description

Specifies the prefix that displays on automatic e-mail notifications.

Access

Public

Syntax

application.instancename.**of_SetNotifyPreTitle** (*prefix*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)
prefix	String specifying the prefix

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetNotifyPreTitle function:

```
...
this.inv_error.of_SetNotifyPreTitle  &
   ("EISAPP AutoMessage - ")
...
```

of_SetNotifySeverity

Description

Specifies the message severity threshold, above which the users specified in the of_SetNotifyWho function are automatically notified via e-mail.

Access

Public

Syntax

application.instancename.**of_SetNotifySeverity** (*threshold*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant. (must be gnv_app).
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
threshold	Integer specifying the value above which the error service sends automatic notification

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

When a message occurs, PFC compares the message's severity with the threshold value. If the message's severity is higher, PFC sends a notification via e-mail.

Examples

This example, from an n_cst_appmanager pfc_Open event, calls the of_SetNotifySeverity function:

```
this.of_SetError(TRUE)
this.inv_error.of_SetPredefinedSource &
   ("c:\eisapp\eiserr.txt")
this.inv_error.of_SetNotifySeverity(4)
```

of_SetNotifyWho

Description

Establishes the list of e-mail user IDs that are notified when certain errors occur.

MAPI-compliant e-mail systems only
PFC's automatic error notification works with MAPI-compliant systems only. To use this feature with non-MAPI e-mail systems, you must extend this functionality.

Access

Public

Syntax

application.instancename.**of_SetNotifyWho** (*userlist* {, *addresslist* })

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
userlist	String array containing the e-mail IDs of users to be notified when certain errors occur
addresslist	String array containing the e-mail addresses of users to be notified when certain errors occur

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetNotifyWho function:

```
String    ls_notify[ ] = {"Doe, John", "Burt, Ben"}

gnv_app.inv_error.of_SetNotifyWho(ls_notify)
```

of_SetPredefinedSource

Description

Specifies the source for predefined messages. This is either a file or a Transaction object connected to a database containing the messages. This function also loads the messages from the source into the ids_messages DataStore.

Access

Public

Syntax

application.instancename.**of_SetPredefinedSource** (*source*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
source	String specifying a filename or n_tr variable for predefined messages. If this is an n_tr variable, it must already be instantiated and connected to the message database

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If *source* specifies an n_tr-based Transaction object, it must be instantiated and the database must contain a table named *messages* whose columns match those in PFC's d_definedmessages DataWindow object.

If *source* specifies a filename, the file must contain tab-delimited messages in a format that corresponds to PFC's d_definedmessages DataWindow object; and the file must contain predefined PFC messages, copied from the messages table in the PFC database.

Examples

This example calls the of_SetPredefinedSource function:

```
String    ls_source
Integer   li_return

ls_source = "c:\eisapp\eismsg.txt"
li_return = &
   gnv_app.inv_error.of_SetPredefinedSource &
      (ls_source)
```

of_SetStyle

Description

Specifies the message box style:

PowerBuilder MessageBox
PFC w_message window

Access

Public

Syntax

application.instancename.**of_SetStyle** (*style*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
style	Integer or n_cst_error constant specifying the message box style: ♦ **DEFAULT or 0** PowerBuilder MessageBox ♦ **PFCWINDOW or 1** PFC w_message window

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If you use the w_message window, the of_Message function can optionally include print and user input capabilities.

Examples

This example calls the of_SetStyle function:

```
Integer   li_return
```

```
li_return = gnv_app.inv_error.of_SetStyle  &
   (gnv_app.inv_error.PFCWINDOW)
...
```

of_SetTimeOut

Description

Specifies the number of seconds a w_message dialog box will stay open.

Access

Public

Syntax

application.instancename.**of_SetTimeOut** (*seconds*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
seconds	Integer specifying the number of seconds that the w_message dialog box will stay open. If you specify 0, the dialog box stays open until the user closes it

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Timeout processing only applies when using the w_message window.

Examples

This example calls the of_SetTimeOut function:

```
...
this.inv_error.of_SetTimeOut(60)
...
```

of_SetUnattended

Description

Specifies whether the unattended option is enabled.

Access

Public

Syntax

application.instancename.**of_SetUnattended** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
boolean	Boolean indicating whether unattended processing is enabled (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Use unattended processing to suppress message display while still using message logging and automatic e-mail notification.

Unattended processing only applies when using the w_message window.

Examples

This example calls the of_SetUnattended function:

```
...
this.inv_error.of_SetLogFile &
   ("c:\pb6i32\errlog.txt")
this.inv_error.of_SetUnattended(TRUE)
...
```

of_SetUser

Description

Specifies the user name.

Access

Public

Syntax

application.instancename.**of_SetUser** ( )

Argument	Description
application	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
instancename	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The user name is used in message logging and automatic e-mail notification.

Examples

This example calls the of_SetUser function:

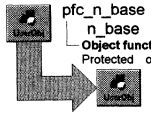
```
this.inv_error.of_SetUser(this.of_GetUserID())
```

n_cst_filesrv

Description

Ancestor object for platform-specific file-access functions. Many of the functions in this object are virtual functions and return -1, FALSE, or an empty string. A virtual function is an empty placeholder function; the actual processing occurs in its descendants. Defining virtual functions in an ancestor object is a technique that allows you to define a reference variable with the ancestor's data type but instantiate the variable with the descendant's data type. PFC contains descendent objects for each supported platform (for example 16-bit windows, 32-bit windows, Macintosh, or UNIX).

Ancestry



Public	of_AssemblePath	Public	of_GetFileSize
Protected	of_CalculateFileAttributes	Public	of_GetLastAccessDate
Public	of_ChangeDirectory	Public	of_GetLastWriteDate
Public	of_CreateDirectory	Public	of_GetLastWriteDateTime
Public	of_DelTree	Public	of_GetLastWriteTime
Protected	of_DirAttribToDS	Public	of_GetLongFilename
Public	of_DirectoryExists	Public	of_GetSeparator
Public	of_Dirlist	Public	of_GetVolumes
Protected	of_DSToDirAttrib	Protected	of_IncludeFile
Public	of_FileCopy	Public	of_ParsePath
Public	of_FileRead	Public	of_RemoveDirectory
Public	of_FileRename	Public	of_SetCreationDateTime
Public	of_FileWrite	Public	of_SetFileArchive
Public	of_GetAltFilename	Public	of_SetFileAttributes
Public	of_GetCreationDate	Public	of_SetFileHidden
Public	of_GetCreationDateTime	Public	of_SetFileReadonly
Public	of_GetCreationTime	Public	of_SetFileSystem
Public	of_GetCurrentDirectory	Public	of_SetLastAccessDate
Public	of_GetDiskSpace	Public	of_SetLastWriteDateTime
Public	of_GetDriveType	Public	of_SortDirlist
Public	of_GetFileAttributes		

Library

PFCAPSRV.PBL
PFEAPSRV.BPL

Object relationships

n_cst_dirattrib
n_cst_numerical
n_cst_string
n_ds

Usage

To use the file service:

1. Instantiate the appropriate file services object by calling the f_SetFilesrv global function. This function examines your environment and instantiates the appropriate user object.
2. Call file services functions, as required. The code for many of these functions is defined in the descendent objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_filesrv) function, which returns -1, FALSE, or an empty string.

Descendants

n_cst_filesrvaix (see n_cst_filesrvsol2)

n_cst_filesrvhpux (see n_cst_filesrvsol2)
n_cst_filesrvmac
n_cst_filesrvsol2
n_cst_filesrvwin16
n_cst_filesrvwin32

See also

f_SetFilesrv
n_cst_platform

Instance variables

N_cst_filesrv includes instance variables:

Instance variable	Description	Data type	Access	Usage
ids_files	Used for sorting	n_ds	Protected	Internal
is_allfiles	Operating system specific characters to indicate all files (for example, *.* in DOS)	String	Protected	Internal
is_separator	Separator character used by the operating system when specifying a file's fully qualified pathname	String	Protected	Internal

Functions

N_cst_filesrv includes precoded object functions, most of which are virtual functions with corresponding versions in n_cst_filesrv descendants. With few exceptions, these functions return -1 or an empty string.

By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

N_cst_filsrv includes precoded object functions:

of_AssemblePath	of_GetFileSize
of_CalculateFileAttributes	of_GetLastAccessDate
of_ChangeDirectory	of_GetLastWriteDate
of_CreateDirectory	of_GetLastWriteDateTime
of_DelTree	of_GetLastWriteTime

of_DirAttribToDS	of_GetLongFilename
of_DirectoryExists	of_GetSeparator
of_DirList	of_GetVolumes
of_DSToDirAttrib	of_IncludeFile
of_FileCopy	of_ParsePath
of_FileRead	of_RemoveDirectory
of_FileRename	of_SetCreationDateTime
of_FileWrite	of_SetFileArchive
of_GetAltFilename	of_SetFileAttributes
of_GetCreationDate	of_SetFileHidden
of_GetCreationDateTime	of_SetFileReadonly
of_GetCreationTime	of_SetFileSystem
of_GetCurrentDirectory	of_SetLastAccessDate
of_GetDiskSpace	of_SetLastWriteDateTime
of_GetDriveType	of_SortDirList
of_GetFileAttributes	

of_AssemblePath

Description

Assembles a fully qualified directory path from its component parts.

Access

Public

Syntax

instancename.**of_AssemblePath** (*drive*, *path*, *filename* {, *extension* })

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	String specifying the drive name. The colon is optional
path	String specifying the directory path. The initial backslash is optional
filename	String specifying the name of the file
extension (optional)	String specifying the 1-to-3 character file extension. The default is an empty string

Return value

String. Returns the fully qualified directory path.

Examples

This example calls the of_AssemblePath function (it assumes an inv_filesrv instance variable):

```
String    ls_drive, ls_dirpath, ls_filename
String    ls_extension, ls_fullpath

ls_fullpath = inv_filesrv.of_AssemblePath &
```

```
(ls_drive, ls_dirpath, ls_filename, ls_extension)
```

of_CalculateFileAttributes

Description

Calculates a file's attribute byte.

Access

Protected

Syntax

instancename.**of_CalculateFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the files whose attributes will be set. If *filename* does not specify a fully qualified path, the function works relative to the current directory
readonly	Boolean indicating the read-only attribute. If you specify NULL for this argument, the function does not change the read-only attribute
hidden	Boolean indicating the hidden attribute. If you specify NULL for this argument, the function does not change the hidden attribute
system	Boolean indicating the system attribute. If you specify NULL for this argument, the function does not change the system attribute
archive	Boolean indicating the archive attribute. If you specify NULL for this argument, the function does not change the archive attribute

Return value

UnsignedLong. Returns the new attribute byte if the function succeeds and -1 if an error occurs.

Examples

This example is from the n_cst_filesrvwin16 of_SetFileAttributes function:

```
Integer   li_rc
ULong     lul_attrib

lul_attrib = of_CalculateFileAttributes &
   (as_filename, ab_readonly, ab_hidden, &
      ab_system, ab_archive)
IF lul_Attrib = -1 THEN Return -1
...
```

of_ChangeDirectory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_ChangeDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_ChangeDirectory function
n_cst_filesrvsol2 of_ChangeDirectory function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_ChangeDirectory function
n_cst_filesrvwin32 of_ChangeDirectory function

of_CreateDirectory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_CreateDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_CreateDirectory function
n_cst_filesrvsol2 of_CreateDirectory function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_CreateDirectory function
n_cst_filesrvwin32 of_CreateDirectory function

of_DelTree

Description

Recursively deletes a directory, along with all of its files and subdirectories.

Access

Public

Syntax

instancename.**of_DelTree** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the path to the directory to be deleted. If you do not specify an absolute path, this function deletes relative to the current working directory

Return value

Integer. Returns 0 if the function succeeds, 1 if the directory does not exist, and a negative number if an error occurs.

Examples

This example erases the directory named in the sle_del_directory SingleLineEdit, its files, its subdirectories, and its subdirectory's files (it assumes an inv_filesrv instance variable):

```
Integer   li_return

li_return = MessageBox("Delete", &
   "Delete all files and directories in " + &
      sle_del_directory.Text + "?", &
         Exclamation!, OKCancel!, 2)
IF li_return = 1 THEN
   li_return = inv_filesrv.of_DelTree &
      (sle_del_directory.Text)
   IF li_return = 1 THEN
      MessageBox("Delete", "Directory not found")
      Return
...
```

of_DirAttribToDS

Description

Copies the passed n_cst_dirattrib structure object to a DataStore.

Access

Protected

Syntax

instancename.**of_DirAttribToDS** (*datastore, entry, file, group*)

Argument	Description
instancename	Instance name of n_cst_filesrv
datastore	N_ds
entry	N_cst_dirattrib
file	String
group	Integer

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SortDirList function:

```
...
If anv_DirList[li_Cnt].ib_Subdirectory Then
  ls_sortfilename = Mid(lnv_direntry.is_FileName, &
    2, (Len(lnv_direntry.is_FileName) - 2))
  of_DirAttribToDS(lds_files, lnv_direntry,  &
    ls_sortfilename, 0)
...
```

of_DirectoryExists

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_DirectoryExists** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String

Return value

Boolean. Always returns FALSE.

See also

n_cst_filesrvmac of_DirectoryExists function
n_cst_filesrvsol2 of_DirectoryExists function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_DirectoryExists function

n_cst_filesrvwin32 of_DirectoryExists function

of_DirList

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_DirList** (*filespec, filetype, dirlist, sorttype*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filespec	String
filetype	Long
dirlist	N_cst_dirlistattrib structure object (passed by reference)
sorttype	Integer

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_DirList function
n_cst_filesrvsol2 of_DirList function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_DirList function
n_cst_filesrvwin32 of_DirList function

of_DSToDirAttrib

Description

Copies the specified row in the passed DataStore to an n_cst_dirattrib structure object.

Access

Protected

Syntax

instancename.**of_DSToDirAttrib** (*datastore, entry, row*)

Argument	Description
instancename	Instance name of n_cst_filesrv
datastore	N_ds
entry	N_cst_dirattrib (passed by reference)

Argument	Description
row	Long

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SortDirList function:

```
...
ll_rowcount = lds_files.RowCount()
FOR li_cnt = 1 TO ll_rowcount
   of_DSToDirAttrib(lds_files, lnv_direntry, li_cnt)
   anv_dirlist[li_pos] = lnv_direntry
   li_pos++
NEXT
...
```

of_FileCopy

Description

Copies one file to another, optionally appending to the target file.

Access

Public

Syntax

instancename.**of_FileCopy** (*sourcefile*, *targetfile* {, *append* })

Argument	Description
instancename	Instance name of n_cst_filesrv
sourcefile	String specifying the name of the file to be copied. If *sourcefile* does not specify a fully qualified path, the function works relative to the current directory
targetfile	String specifying the name of the target file. If *targetfile* does not specify a fully qualified path, the function works relative to the current directory
append (optional)	Boolean specifying whether to append to *targetfile* (TRUE) or replace it (FALSE)

Return value

Integer. Returns values as follows:

0—Success
-1—Error opening *sourcefile*
-2—Error writing *targetfile*

Examples

This example calls the of_FileCopy function to copy the file specified in the sle_source SingleLineEdit to the file specified in the sle_target SingleLineEdit, replacing the previous contents of the target file. It assumes that n_cst_filesrv has been created using the inv_filesrv instance variable:

```
Integer   li_return

li_return = &
   inv_filesrv.of_FileCopy(sle_source.text, &
      sle_target.text, FALSE)
CHOOSE CASE li_return
   CASE -1
      MessageBox("Copy Error", &
         "Error opening source file")
   CASE -2
      MessageBox("Copy Error", &
         "Error reading source file")
   CASE -3
      MessageBox("Copy Error", &
         "Error accessing target file")
END CHOOSE
```

of_FileRead

Reads a file into a Blob or string array. There are two syntaxes:

To	Use
Read a file into a BLOB variable	Syntax 1
Read a file into a string array	Syntax 2

Syntax 1

To read a file into a Blob variable

Description

Reads a file into a Blob variable.

Access

Public

Syntax

instancename.**of_FileRead** (*filename*, *blob*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file to be read. If *filename* does not specify a fully qualified path, the function works relative to the current directory

Argument	Description
blob	Blob variable into which the function copies the contents of *filename* (passed by reference)

Return value

Long. Returns the size of *blob* if the function succeeds and -1 if an error occurs.

Usage

Call this function to access files in any size or format, including bitmaps and text files.

Use Syntax 2 to read a file into a string array.

Examples

This example calls the of_FileRead function (it assumes an inv_filesrv instance variable):

```
Long   ll_return
Blob   lblb_file

ll_return = inv_filesrv.of_FileRead &
   (sle_filename.text, lblb_file)
CHOOSE CASE ll_return
   CASE -1
      MessageBox("Error", "Error accessing file")
   CASE ELSE
      // File processing goes here
END CHOOSE
```

Syntax 2

To read a file into a string array

Description

Reads the contents of a file into a string array. Each element in the string array holds 32K of data.

Access

Public

Syntax

instancename.**of_FileRead** (*filename*, *text*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file to be read. If *filename* does not specify a fully qualified path, the function works relative to the current directory
text	String array into which the function copies the contents of *filename*

Return value

Integer. Returns the number of elements in the *text* array if the function succeeds and -1 if an error occurs.

Usage

Call this function to access text files of any size.

Use Syntax 1 to read files of any format into a Blob.

Examples

This example calls the of_FileRead function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_file[ ]

li_return = inv_filesrv.of_FileRead &
   (sle_filename.text, ls_file)
CHOOSE CASE li_return
   CASE -1
      MessageBox("Error", "Error accessing file")
   CASE ELSE
      // File processing goes here
END CHOOSE
```

of_FileRename

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_FileRename** (*oldname*, *newname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
oldname	String
newname	String

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_FileRename function
n_cst_filesrvsol2 of_FileRename function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_FileRename function
n_cst_filesrvwin32 of_FileRename function

of_FileWrite

Writes a file from a Blob or string. There are two syntaxes:

To	Use
Write a file from a Blob variable	Syntax 1
Write a file from a string	Syntax 2

Syntax 1

To write from a BLOB variable

Description

Writes the contents of a blob variable to a file.

Access

Public

Syntax

instancename.**of_FileWrite** (*filename*, *blob* {, *append* })

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file to be written. If *filename* does not specify a fully qualified path, the function works relative to the current directory
blob	Blob variable from which the function copies to *filename*
append (optional)	Boolean specifying whether to append to (TRUE) or overwrite (FALSE) *filename*. The default is TRUE

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to write files in any format, including bitmaps and text files.

Examples

This example calls the of_FileWrite. It assumes that n_cst_filesrv has been created using the inv_filesrv instance variable:

```
Blob   lblb_file
... // Logic to populate lblb_file
IF inv_filesrv.of_FileWrite &
   (sle_filename.text, lblb_file, FALSE) = -1 THEN
      MessageBox("Write Error", &
         "Error writing BLOB file")
END IF
```

Syntax 2

To write from a string

Description

Writes to a text file using the contents of a passed string.

Access

Public

Syntax

instancename.**of_FileWrite** (*filename*, *text* {, *append* })

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file to be written. If *filename* does not specify a fully qualified path, the function works relative to the current directory
text	String from which the function copies to *filename*
append (optional)	Boolean specifying whether to append to (TRUE) or overwrite (FALSE) *filename*. The default is TRUE

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_FileWrite function (it assumes an inv_filesrv instance variable):

```
String    ls_file
... // Logic to populate ls_file
IF inv_filesrv.of_FileWrite &
   (sle_filename.text, ls_file, FALSE) = -1 THEN
      MessageBox("Write Error", &
         "Error writing file")
END IF
```

of_GetAltFilename

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetAltFilename** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String

Return value

String. Always returns an empty string.

Usage

This function is not available on 16-bit Windows.

See also

n_cst_filesrvwin32 of_GetAltFilename function
n_cst_filesrvsol2 of_GetAltFilename function

(also in n_cst_filesrvaix and n_cst_filsrv_hpux)

of_GetCreationDate

Description

Retrieves the date the file was created.

Access

Public

Syntax

instancename.**of_GetCreationDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose creation date is to be returned. If *filename* does not specify a fully qualified path, the function works relative to the current directory
date	Date variable into which the function places the date *filename* was created (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if a problem occurs.

Usage

This function calls the of_GetCreationDateTime function. It is not available on 16-bit Windows.

Examples

This example calls the of_GetCreationDate function (it assumes an inv_filesrv instance variable):

```
Date   ldt_date

inv_filesrv.of_GetCreationDate &
   (sle_filename.Text, ldt_date)
MessageBox("Creation Date", &
   "Creation date: " + String(ldt_date) )
```

of_GetCreationDateTime

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetCreationDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String
date	Date
time	Time

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_GetCreationDateTime function
n_cst_filesrvsol2 of_GetCreationDateTime function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin32 of_GetCreationDateTime function

of_GetCreationTime

Description

Retrieves the time the file was created.

Access

Public

Syntax

instancename.**of_GetCreationTime** (*filename*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose creation time is to be returned. If *filename* does not specify a fully qualified path, the function works relative to the current directory
time	Time variable into which the function places the time *filename* was created (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if a problem occurs.

Usage

This function calls the of_GetCreationDateTime function. It is not available on 16-bit Windows.

Examples

This example calls the of_GetCreationTime function (it assumes an inv_filesrv instance variable):

```
Time   lt_time
```

```
inv_filesrv.of_GetCreationTime &
   (sle_filename.Text, lt_time)
MessageBox("Creation Time", &
   "Creation time: " + String(lt_time) )
```

of_GetCurrentDirectory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetCurrentDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_filesrv

Return value

String. Always returns an empty string.

See also

n_cst_filesrvmac of_GetCurrentDirectory function
n_cst_filesrvwin16 of_GetCurrentDirectory function
n_cst_filesrvsol2 of_GetCurrentDirectory function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin32 of_GetCurrentDirectory function

of_GetDiskSpace

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetDiskSpace** (*drive*, *total*, *free*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character
total	Long
free	Long

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_GetDiskSpace function
n_cst_filesrvsol2 of_GetDiskSpace function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_GetDiskSpace function
n_cst_filesrvwin32 of_GetDiskSpace function

of_GetDriveType

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetDriveType** (*drive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_GetDriveType function
n_cst_filesrvsol2 of_GetDriveType function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_GetDriveType function
n_cst_filesrvwin32 of_GetDriveType function

of_GetFileAttributes

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetFileAttributes** (*filename, readonly, hidden, system, subdirectory, archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv

Argument	Description
filename	String
readonly	Boolean
hidden	Boolean
system	Boolean
subdirectory	Boolean
archive	Boolean

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_GetFileAttributes function
n_cst_filesrvsol2 of_GetFileAttributes function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_GetFileAttributes function
n_cst_filesrvwin32 of_GetFileAttributes function

of_GetFileSize

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetFileSize** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String

Return value

Double. Always returns -1.

See also

n_cst_filesrvmac of_GetFileSize function
n_cst_filesrvsol2 of_GetFileSize function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_GetFileSize function
n_cst_filesrvwin32 of_GetFileSize function

of_GetLastAccessDate

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetLastAccessDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose last access date is to be returned. If *filename* does not specify a fully qualified path, the function works relative to the current directory
date	Date variable into which the function places the date *filename* last accessed (passed by reference)

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_GetLastAccessDate function
n_cst_filesrvsol2 of_GetLastAccessDate function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin32 of_GetLastAccessDate function

of_GetLastWriteDate

Description

Retrieves the date the file was last modified.

Access

Public

Syntax

instancename.**of_GetLastWriteDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose last write date is to be returned. If *filename* does not specify a fully qualified path, the function works relative to the current directory
date	Date variable into which the function places the date *filename* last modified (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if a problem occurs.

Usage

This function calls the of_GetLastWriteDateTime function.

Examples

This example calls the of_GetLastWriteDate function (it assumes an inv_filesrv instance variable):

```
Date   ldt_date

inv_filesrv.of_GetLastWriteDate &
   (sle_filename.Text, ldt_date)
MessageBox("Last Write Date", &
   "Last write date: " + String(ldt_date) )
```

of_GetLastWriteDateTime

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String
date	Date
time	Time

Return value

Integer. Always returns -1.

See also

n_cst_filesrvwin16 of_GetLastWriteDateTime function
n_cst_filesrvsol2 of_GetLastWriteDateTime function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin32 of_GetLastWriteDateTime function

of_GetLastWriteTime

Description

Retrieves the time the file was last modified.

Access

Public

Syntax

instancename.**of_GetLastWriteTime** (*filename*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose last write date is to be returned. If *filename* does not specify a fully qualified path, the function works relative to the current directory
time	Time variable into which the function place the time *filename* last modified (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the of_GetLastWriteDateTime function.

Examples

This example calls the of_GetLastWriteTime function (it assumes an inv_filesrv instance variable):

```
Time   lt_time

inv_filesrv.of_GetLastWriteTime &
   (sle_filename.Text, lt_time)
MessageBox("Last Write Time", &
   "Last write time: " + String(lt_time) )
```

of_GetLongFilename

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_GetLongFilename** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String

Return value

String. Always returns an empty string.

Usage

This function is not available on 16-bit Windows.

See also

n_cst_filesrvwin32 of_GetLongFilename function
n_cst_filesrvsol2 of_GetLongFilename function

(also in n_cst_filesrvaix and n_cst_filsrv_hpux)

of_GetSeparator

Description: Retrieves the directory separator character.

Access: Public

Syntax: *instancename*.**of_GetSeparator** ( )

Argument	Description
instancename	Instance name of n_cst_filesrv

Return value: String. Returns the is_separator instance variable.

Usage: N_cst_filesrv descendants can call this function to create fully qualified pathnames.

Examples: This example calls the of_GetSeparator function:

```
String    is_separator

is_separator = inv_filesrv.of_GetSeparator( )
```

of_GetVolumes

Description: Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access: Public

Syntax: *instancename*.**of_GetVolumes** (*volumes*)

Argument	Description
instancename	Instance name of n_cst_filesrv
volumes	String

Return value: Integer. Always returns -1.

Usage: This function is available on the Macintosh only.

See also: n_cst_filesrvmac of_GetVolumes function

of_IncludeFile

Description

Called by the of_DirList function to report whether a file contains the specified attribute settings.

Access

Protected

Syntax

instancename.**of_IncludeFile** (*filename, attributemask, fileattrib*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file to be accessed. If *filename* does not specify a fully qualified path, the function works relative to the current directory
attributemask	Long containing the bit string that determines whether *filename* is included by of_DirList
fileattrib	UnsignedLong containing the file's attribute bits

Return value

Boolean. Returns TRUE if *filename* should be included and FALSE if it should not.

Usage

The of_DirList function calls this function to determine whether to include the file in its list of files.

The n_cst_filesrvmac of_IncludeFile function overrides this function when executing on the Macintosh.

Examples

This example is from the of_DirList function:

```
...
DO WHILE li_RC = 0
   lui_Attrib = Asc(lstr_FindData.c_Attrib)

   // Determine if this file should be included.
   IF of_IncludeFile(String(lstr_FindData.c_Name), &
      al_FileType, lui_Attrib) THEN
...
```

of_ParsePath

Description

Separates a fully qualified filename into its component parts.

Access

Public

Syntax

instancename.**of_ParsePath** (*path, drive, directorypath, filename* {, *extension* })

Argument	Description
instancename	Instance name of n_cst_filesrv
path	String specifying the fully qualified filename
drive	String into which the function places the drive specification, including the colon and backslash (if specified in *fullname*) (passed by reference)
directorypath	String into which the function places the directory path (passed by reference)
filename	String into which the function places the filename without the extension (passed by reference)
extension (optional)	String into which the function places the extension (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_ParsePath function (it assumes an inv_filesrv instance variable):

```
String    ls_drive, ls_path, ls_filename
String    ls_extension

inv_filesrv.of_ParsePath &
   (sle_fullpath.Text, ls_drive, ls_path, &
      ls_filename, ls_extension)
sle_filename.Text = ls_filename + "." + ls_extension
```

of_RemoveDirectory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_RemoveDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv

Argument	Description
directoryname	String

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_RemoveDirectory function
n_cst_filesrvsol2 of_RemoveDirectory function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin16 of_RemoveDirectory function
n_cst_filesrvwin32 of_RemoveDirectory function

of_SetCreationDateTime

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_SetCreationDateTime** (*filename, date, time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String
date	Date
time	Time

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_SetCreationDateTime function
n_cst_filesrvwin32 of_SetCreationDateTime function
n_cst_filesrvsol2 of_SetCreationDateTime function

(also in n_cst_filesrvaix and n_cst_filsrv_hpux)

of_SetFileArchive

Description

Calls the of_SetFileAttributes function to set a file's archive attribute.

Access

Public

Syntax

instancename.**of_SetFileArchive** (*filename, archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose archive attribute is to be set. If *filename* does not specify a fully qualified path, the function works relative to the current directory
archive	Boolean specifying whether to turn the archive bit on (TRUE) or off (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileArchive function (it assumes an inv_filesrv instance variable):

```
inv_filesrv.of_SetFileArchive &
    (sle_filename.Text, cbx_archive.Checked)
```

of_SetFileAttributes

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_SetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String
readonly	Boolean
hidden	Boolean
system	Boolean
archive	Boolean

Return value

Integer. Always returns -1.

See also

n_cst_filesrvmac of_SetFileAttributes function
n_cst_filesrvsol2 of_SetFileAttributes function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)

n_cst_filesrvwin16 of_SetFileAttributes function
n_cst_filesrvwin32 of_SetFileAttributes function

of_SetFileHidden

Description

Calls the of_SetFileAttributes function to set a file's hidden attribute.

Access

Public

Syntax

instancename.**of_SetFileHidden** (*filename*, *hidden*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose hidden attribute is to be set. If *filename* does not specify a fully qualified path, the function works relative to the current directory
hidden	Boolean specifying whether to make the file hidden (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileHidden function (it assumes an inv_filesrv instance variable):

```
inv_filesrv.of_SetFileHidden &
    (sle_filename.Text, cbx_hidden.Checked)
```

of_SetFileReadonly

Description

Calls the of_SetFileAttributes function to set a file's read-only attribute.

Access

Public

Syntax

instancename.**of_SetFileReadonly** (*filename*, *readonly*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose read-only attribute is to be set. If *filename* does not specify a fully qualified path, the function works relative to the current directory
readonly	Boolean specifying whether to make the file read-only (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileReadonly function (it assumes an inv_filesrv instance variable):

```
inv_filesrv.of_SetFileReadonly &
    (sle_filename.Text, cbx_readonly.Checked)
```

of_SetFileSystem

Description

Calls the of_SetFileAttributes function to set a file's system attribute.

Access

Public

Syntax

instancename.**of_SetFileSystem** (*filename*, *system*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the name of the file whose system attribute is to be set. If *filename* does not specify a fully qualified path, the function works relative to the current directory
system	Boolean specifying whether to set the file's system attribute (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileSystem function (it assumes an inv_filesrv instance variable):

```
inv_filesrv.of_SetFileSystem &
    (sle_filename.Text, cbx_system.Checked)
```

of_SetLastAccessDate

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_SetLastAccessDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String
date	Date

Return value

Integer. Always returns -1.

Usage

This function does not run on 16-bit Windows.

See also

n_cst_filesrvmac of_SetLastAccessDate function
n_cst_filesrvsol2 of_SetLastAccessDate function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin32 of_SetLastAccessDate function

of_SetLastWriteDateTime

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_filesrv for use with all n_cst_filesrv descendants.

Access

Public

Syntax

instancename.**of_SetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String
date	Date
time	Time

Return value

Integer. Always returns -1.

See also

n_cst_filesrvwin16 of_SetLastWriteDateTime function
n_cst_filesrvsol2 of_SetLastWriteDateTime function (also in n_cst_filesrvaix and n_cst_filsrv_hpux)
n_cst_filesrvwin32 of_SetLastWriteDateTime function

of_SortDirList

Description

Sorts the directory list output from the of_DirList function.

Access

Public

Syntax

instancename.**of_SortDirList** (*dirlist*, *sorttype* {, *ascending* })

Argument	Description
instancename	Instance name of n_cst_filesrv
dirlist	Array of type n_cst_dirattrib containing the output from the of_DirList function (passed by reference)
sorttype	Integer specifying how to sort *dirlist*: ♦ 1 By filename ♦ 2 By extension ♦ 3 By last write date/time ♦ 4 By filesize
ascending (optional)	Boolean indicating whether the sort direction is ascending (TRUE) or descending (FALSE). The default is TRUE

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The of_DirList function calls this function to sort the *dirlist* array.

Examples

This example calls the of_SortDirList function:

```
String     ls_currdir
Integer    li_cnt, li_entries
String     ls_import
n_cst_dirlistattrib lnv_dirlist[ ]

SetPointer(HourGlass!)
ls_currdir = "c:\Win95"

li_entries = &
   inv_filesrv.of_DirList &
      (ls_currdir, 0, lnv_dirlist)
...
inv_filesrv.of_SortDirList(lnv_dirlist, 1, TRUE)
...
```

n_cst_filesrvmac

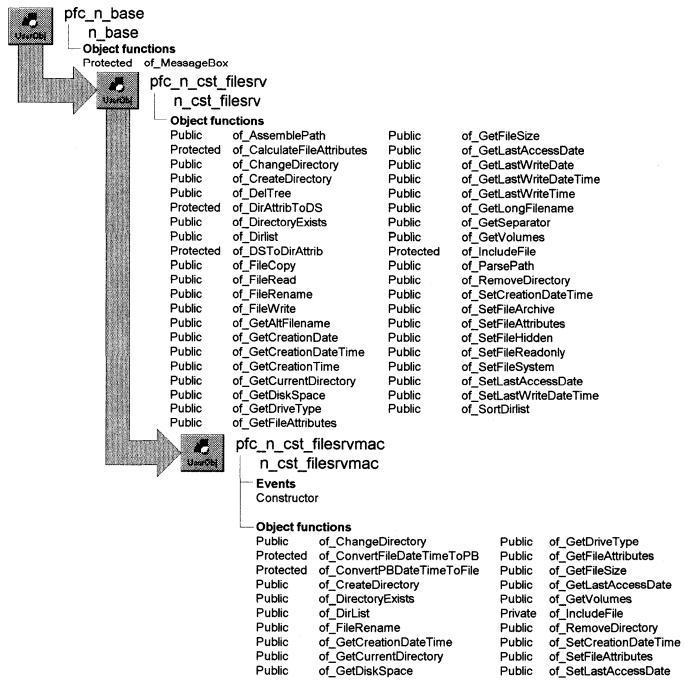
Description

Platform-specific file management functions for applications running on the Macintosh. Platform-specific file management functions require a call to a platform-specific external function. When you call the f_SetFilesrv global function, it detects the operating environment and instantiates the correct platform-specific object.

Many of the functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar.

This object calls functions in the platform-specific module PFCMacFileSrv.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_dirattrib
n_cst_numerical

Usage

To use the file management service:

1 Instantiate the appropriate file management object by calling the f_SetFilesrv global function. This function examines your environment and instantiates the appropriate user object.

2 Call file management functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_filesrv) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_finddata	s_filename	String
	ul_creationtime	UnsignedLong
	ul_fileattributes	UnsignedLong
	ul_filesize	UnsignedLong
	ul_lastaccesstime	UnsignedLong
os_systemtime	ui_wyear	Long
	ui_wday	Long
	ui_wdayofweek	Long
	ui_whour	Long
	ui_milliseconds	Long
	ui_wminute	Long
	ui_wmonth	Long
	ui_wsecond	Long

See also

f_SetFilesrv
n_cst_filesrv
n_cst_filesrvsol2
n_cst_filesrvwin16
n_cst_filesrvwin32
n_cst_platform

Events

N_cst_filesrvmac contains a precoded event:

Constructor

Constructor

Description — Initializes the is_separator and is_allfiles instance variables.

Usage — This event executes when the object is created.

Functions

N_cst_filesrvmac includes precoded object functions:

of_ChangeDirectory	of_GetDriveType
of_ConvertFileDateTimeToPB	of_GetFileAttributes
of_ConvertPBDateTimeToFile	of_GetFileSize
of_CreateDirectory	of_GetLastAccessDate
of_DirectoryExists	of_GetVolumes
of_DirList	of_IncludeFile
of_FileRename	of_RemoveDirectory
of_GetCreationDateTime	of_SetCreationDateTime
of_GetCurrentDirectory	of_SetFileAttributes
of_GetDiskSpace	of_SetLastAccessDate

of_ChangeDirectory

Description — Changes the current working folder.

Access — Public

Syntax — *instancename*.**of_ChangeDirectory** (*foldername*)

Argument	Description
instancename	Instance name of n_cst_filesrv
foldername	String specifying the name of the new working folder. If you do not specify an absolute path, this function changes folders relative to the current working folder

Return value — Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If your application performs functions that change the folder, call this function before path-sensitive PowerScript functions, such as ShowHelp, to ensure access to the proper files.

This function calls the SetCurrentDirectory external function.

Examples

This example changes folders to the HDMAC:realapp folder before calling the SetProfileString function (it assumes an inv_filesrv instance variable):

```
Integer li_return
String ls_language

ls_language = sle_curr_language.Text
li_return = inv_filesrv.of_ChangeDirectory &
   ("HDMAC:realapp")
IF li_return = 1 THEN
   SetProfileString("realapp.ini", "language", &
      "currentlanguage", ls_language)
END IF
```

of_ConvertFileDateTimeToPB

Description

Converts a file date (UnsignedLong) into a PowerBuilder date.

Access

Protected

Syntax

instancename.**of_ConvertFileDateTimeToPB** (*filedatetime*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filedatetime	UnsignedLong containing a file's system date and time
date	Date variable into which the function places the date (passed by reference)
time	Time variable into which the function places the time (passed by reference)

Return value

Date. Returns *date* and *time* in PowerBuilder format.

Usage

Internal.

This function calls the GetSysTimeFromFileTime external function.

Examples

This example is from the of_DirList function:

```
Integer   li_entries
Long      ll_cnt
String    ls_time, ls_date
String    ls_drive
Boolean   lb_found
Time      lt_time
os_finddata       lstr_finddata[]
n_cst_dirattrib   lnv_empty[]
n_cst_numerical   lnv_numeric
Long      ll_count, ll_i
String    ls_filespec
Long      ll_pos_temp, ll_pos=0
...
of_ConvertFileDatetimeToPB &
    (lstr_FindData[ll_i].ul_CreationTime, &
       anv_DirList[li_Entries].id_CreationDate, &
          anv_DirList[li_Entries].it_CreationTime)
...
```

of_ConvertPBDateTimeToFile

Description

Converts a PowerBuilder date and time to a system date and time.

Access

Protected

Syntax

instancename.**of_ConvertPBDateTimeToFile** (*date*, *time*, *datetime*)

Argument	Description
instancename	Instance name of n_cst_filesrv
date	Date variable containing a date to be converted
time	Time variable containing a time to be converted
datetime	UnsignedLong into which the function places the converted system date and time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

This function calls the GetFileTimeFromSysTime external function.

Examples

This example is from the of_SetCreationDateTime function:

```
Boolean   lb_ret
```

```
ulong     lul_filetime
os_systemtime   lstr_createsystemtime
os_systemtime   lstr_modsystemtime

If of_ConvertPBDatetimeToFile(ad_filedate, &
   at_filetime, lul_filetime) < 0 Then Return -1
...
```

of_CreateDirectory

Description: Creates a new folder.

Access: Public

Syntax: *instancename*.**of_CreateDirectory** (*foldername*)

Argument	Description
instancename	Instance name of n_cst_filesrv
foldername	String specifying the name of the new folder. If you do not specify an absolute path, this function creates a folder relative to the current working folder

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: This function calls the CreateDirectory external function.

Examples: This example calls the of_CreateDirectory function to create a temporary folder (it assumes an inv_filesrv instance variable):

```
IF inv_filesrv.of_CreateDirectory &
   ("temp") <> 0 THEN
      MessageBox("File Services", &
         "Error creating temp folder")
END IF
```

of_DirectoryExists

Description: Reports whether the specified folder exists.

Access: Public

Syntax: *instancename*.**of_DirectoryExists** (*foldername*)

Argument	Description
instancename	Instance name of n_cst_filesrv
foldername	String specifying the name of the folder. If you do not specify an absolute path, this function looks for a folder relative to the current working folder

Return value

Boolean. Returns TRUE if the folder exists and FALSE if it does not.

Usage

This function calls the GetFileAttributes external function.

Examples

This example calls the of_DirectoryExists function to check for the presence of a temporary folder. If the directory doesn't exist, the example creates the temporary folder (it assumes an inv_filesrv instance variable):

```
Integer   li_return

IF NOT inv_filesrv.of_DirectoryExists &
    ("temp") THEN
       li_return = inv_filesrv.of_CreateDirectory &
          ("temp")
END IF
```

of_DirList

Description

Retrieves information on all files within the passed folder that match the passed arguments.

Access

Public

Syntax

instancename.**of_DirList** (*filespec*, *filetype*, *dirlist*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filespec	String specifying the path and file specification (including wildcards) to the folder containing the files to be listed. If you do not specify an absolute path, this function searches relative to the current working folder

Argument	Description
filetype	Long specifying one or more types of files to return. Types are: ♦ 0—Read/write files ♦ 1—Read-only files ♦ 2—Hidden files ♦ 4—System files ♦ 16—Subdirectories ♦ 32—Archive (modified) files ♦ 16384—Drives ♦ 32768—Exclude read/write files from the list
dirlist	Array of type n_cst_dirattrib into which the function places information for the requested files (passed by reference)

Return value

Integer. Returns the number of elements in the *dirlist* array if the function succeeds and -1 if an error occurs.

Usage

To further sort the *dirlist* array, use the of_SortDirList function.

To list several types, add the *filetype* values together. For example, to list read/write files, subdirectories, and drives, specify 0+16+16384 or 16400 for *filetype*.

This function calls the GetDirList, GetVolumeCount, and GetVolumeName external functions.

Examples

This example creates a list of all system, hidden, read-only, and read/write files in the HDMac:pbapp folder. The function sorts the list by filename and displays it in the dw_dirlist DataWindow (it assumes an inv_filesrv instance variable):

```
String     ls_currdir
Integer    li_cnt, li_entries
String     ls_import
n_cst_dirattrib   lnv_dirlist[ ]

SetPointer(HourGlass!)
ls_currdir = "HDMac:pbapp:*.*"

li_entries = &
   inv_filesrv.of_DirList &
   (ls_currdir, 0, lnv_dirlist)
IF li_Entries < 0 THEN
```

```
    MessageBox("File Services", &
       "Folder not found")
    Return
ELSEIF li_Entries = 0 THEN
    MessageBox("File Services", "No files found")
    Return
END IF

Parent.SetRedraw(False)
mle_dirlist.text = ""
...
```

of_FileRename

Description

Renames or moves a file or folder.

Access

Public

Syntax

instancename.**of_FileRename** (*oldname*, *newname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
oldname	String specifying the complete path of the file to be renamed
newname	String specifying the complete path of the new filename

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the RenameFile external function.

Examples

This example calls the of_FileRename function:

```
String    ls_source, ls_target
Integer   li_return

ls_source = sle_source.text
ls_target = sle_target.text
li_return = inv_filesrv.of_FileRename &
   (ls_source, ls_target)
IF li_return = 1 THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      (ls_source + " renamed to " + ls_target)
ELSE
   MessageBox("File Service", &
```

```
            "Error renaming " + ls_source + " to " &
               + ls_target)
END IF
```

of_GetCreationDateTime

Description

Retrieves the date and time a file was created.

Access

Public

Syntax

instancename.**of_GetCreationDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose creation date and time are to be returned. If you do not specify an absolute path, this function searches relative to the current working folder
date	Date into which the function places the creation date (passed by reference)
time	Time into which the function places the creation time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetFileTime external function.

Examples

This example calls the of_GetCreationDateTime function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_filename
Date      ldt_date
Time      lt_time

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetCreationDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + "~r~nCreation date: " &
```

```
        + String(ldt_date) &
        + "~r~nCreation time: " &
        + String(lt_time) )
END IF
```

of_GetCurrentDirectory

Description

Retrieves the current working folder.

Access

Public

Syntax

instancename.**of_GetCurrentDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_filesrv

Return value

String. Returns the fully qualified path of the current working folder.

Usage

This function calls the GetCurrentDirectory external function.

Examples

This example calls the of_GetCurrentDirectory function and, if the result isn't equal to the is_app_dir instance variable, calls the of_ChangeDirectory function to reset the working folder (it assumes an inv_filesrv instance variable):

```
String   ls_return

ls_return = inv_filesrv.of_GetCurrentDirectory( )
IF ls_return <> is_app_dir THEN
   inv_filesrv.of_ChangeDirectory(is_app_dir)
END IF
```

of_GetDiskSpace

Description

Retrieves space information for a specified disk drive.

Access

Public

Syntax

instancename.**of_GetDiskSpace** (*drive, total, free*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter

Argument	Description
total	Long into which the function returns the total bytes on *drive* (passed by reference)
free	Long into which the function returns the free space on *drive* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to determine the total space and free space for a disk drive.

This function calls the GetDiskFreeSpace external function.

Examples

This example calls the of_GetDiskSpace function:

```
Character   lc_drive
Long        ll_total, ll_free
Integer     li_return

lc_drive = lb_drive.SelectedItem()
li_return = inv_filesrv.of_GetDiskSpace &
   (lc_drive, ll_total, ll_free)
IF li_return = 1 THEN
   MessageBox("File Service", "Drive " + lc_drive &
   + "~r~n" + String(ll_total) + " total bytes~r~n" &
   + String(ll_free) + " total free space")
ELSE
   MessageBox("File Service", &
      "Error in of_GetDiskSpace")
END IF
```

of_GetDriveType

Description

Retrieves the drive type for a specified drive.

Access

Public

Syntax

instancename.**of_GetDriveType** (*drive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter

Return value

Integer. Returns values as follows:

0 Undetermined drive type
1 Root directory does not exist
2 Removable disk
3 Nonremovable disk
4 Remote (network) drive
5 CD-ROM drive
6 RAM disk

Usage

This function calls the GetDriveType external function.

Examples

This example calls the of_GetDriveType function (it assumes an inv_filesrv instance variable):

```
Integer   lui_drivetype
String    ls_driveletter
String    ls_drivetype[ ] = &
   {"Undetermined","Does not exist","Removeable", &
      "Nonremoveable", "Network drive", "CD-ROM", &
         "RAM disk" }

ls_driveletter = lb_drivelist.SelectedItem( )
lui_drivetype = inv_filesrv.of_GetDriveType &
   (ls_driveletter)
MessageBox("File Service", "Drive type is: " &
   + ls_drivetype[lui_drivetype + 1] )
```

of_GetFileAttributes

Description

Retrieves all attributes for a specified file.

Access

Public

Syntax

instancename.**of_GetFileAttributes** (*filename, readonly, hidden, system, subfolder, archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose attributes are to be returned. If you do not specify an absolute path, this function searches relative to the current working folder
readonly	Boolean indicating whether *filename* is read-only (passed by reference)

Argument	Description
hidden	Boolean indicating whether *filename* is hidden (passed by reference)
system	Boolean indicating whether *filename* is a system file (passed by reference)
subfolder	Boolean indicating whether *filename* is a subfolder (passed by reference)
archive	Boolean indicating whether *filename* has the archive bit set (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetFileAttributes external function.

Examples

This example calls the of_GetFileAttributes function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_subdirectory.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
END IF
```

of_GetFileSize

Description

Retrieves the size of a specified file.

Access

Public

Syntax

instancename.**of_GetFileSize** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv

Argument	Description
filename	String specifying the file whose size is to be returned. If you do not specify an absolute path, this function searches relative to the current working folder

Return value

Double. Returns the number of bytes in *filename* if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetFileSize external function.

Examples

This example calls the of_GetFileSize function (it assumes an inv_filesrv instance variable):

```
Double    ld_return
String    ls_filename

ls_filename = sle_source.text
ld_return = &
   inv_filesrv.of_GetFileSize(ls_filename)
IF ld_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a size of " + String(ld_return) &
         + " bytes.")
END IF
```

of_GetLastAccessDate

Description

Retrieves the date a file was last accessed.

Access

Public

Syntax

instancename.**of_GetLastAccessDate** (*filename, date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last access date is to be returned. If you do not specify an absolute path, this function searches relative to the current working folder

Argument	Description
date	Date into which the function places the last access date (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetFileTime external function.

Examples

This example calls the of_GetLastAccessDate function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_filename
Date      ldt_date

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetLastAccessDate &
   (ls_filename, ldt_date)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a last access date of " &
         + String(ldt_date) )
END IF
```

of_GetVolumes

Description

Retrieves a list of the volumes attached to the user's workstation.

Access

Public

Syntax

instancename.**of_GetVolumes** (*volumes*)

Argument	Description
instancename	Instance name of n_cst_filesrv
volumes	String array into which the function places the volumes attached to the user's workstation (passed by reference)

Return value

Integer. Always returns the number of elements in the *volumes* array if the function succeeds and 0 if there are no volumes attached to the workstation.

Usage

This function calls the GetVolumeCount and GetVolumeName external functions.

Examples

This example calls the of_GetVolumes function (it assumes an inv_filesrv instance variable):

```
String    ls_volumes[]
Integer   li_return, li_count

li_return = inv_filesrv.of_GetVolumes &
   (ls_volumes)
FOR li_count = 1 to li_return
   lb_volumes.AddItem(ls_volumes[li_return])
NEXT
...
```

of_IncludeFile

Description

Called by the of_DirList function to determines whether a file contains the specified attribute settings.

Access

Protected

Syntax

instancename.**of_IncludeFile** (*filename, attributemask, fileattrib*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file to be accessed. If *filename* does not specify a fully qualified path, the function works relative to the current directory
attributemask	Long containing the bit string that determines whether *filename* is included by of_DirList
fileattrib	UnsignedLong containing the file's attribute bits

Return value

Boolean. Returns TRUE if *filename* should be included and FALSE if it should not.

Usage

Internal.

Examples

This example is from the of_DirList function:

```
...
FOR ll_i = 1 to ll_count
```

```
    IF of_IncludeFile &
       (String(lstr_finddata[ll_i].s_filename), &
          al_filetype, &
       lstr_finddata[ll_i].ul_fileattributes) THEN
...
```

of_RemoveDirectory

Description

Deletes the specified folder (the folder must be empty). To delete a folder and its files, call of_DelTree.

Access

Public

Syntax

instancename.**of_RemoveDirectory** (*foldername*)

Argument	Description
instancename	Instance name of n_cst_filesrvmac
directoryname	String specifying the path to the folder to be deleted. If you do not specify an absolute path, this function deletes relative to the current working folder

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the RemoveDirectory external function.

Examples

This example checks if pbapp is the current folder and, if it is, deletes the pbapp:temp folder (it assumes an inv_filesrv instance variable):

```
String    ls_return, ls_temp
Integer   li_return

ls_temp = "temp"
ls_return = &
   gnv_app.inv_platform.of_GetCurrentDirectory( )
IF Mid(ls_return, 4, 5) = "pbapp" THEN
   li_return = inv_filesrv.of_RemoveDirectory &
      (ls_temp)
   IF li_return = 1 THEN
      MessageBox("Delete", &
         "Temp folder does not exist")
...
```

of_SetCreationDateTime

Description: Sets a file's creation date and time.

Access: Public

Syntax: *instancename*.**of_SetCreationDateTime** (*filename, date, time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose creation date and time are to be modified. If you do not specify an absolute path, this function searches relative to the current working folder
date	Date variable containing the new creation date
time	Time variable containing the new creation time

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples: This example calls the of_SetCreationDateTime function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_filename
Date      ldt_date
Time      lt_time

ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesrv.of_SetCreationDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
...
```

of_SetFileAttributes

Description: Sets attributes for a file.

Access: Public

Syntax

instancename.**of_SetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrvmac
filename	String specifying the file whose attributes are to be set. If you do not specify an absolute path, this function searches relative to the current working folder
readonly	Boolean indicating whether *filename* is read-only. If you specify NULL, the function doesn't modify this attribute
hidden	Boolean indicating whether *filename* is hidden. If you specify NULL, the function doesn't modify this attribute
system	Boolean indicating whether *filename* is a system file. If you specify NULL, the function doesn't modify this attribute
archive	Boolean indicating whether *filename* has the archive bit set. If you specify NULL, the function doesn't modify this attribute

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileAttributes function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_SetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
END IF
```

of_SetLastAccessDate

Description

Sets a file's last-accessed date and time.

Access

Public

Syntax

instancename.**of_SetLastAccessDate** (*filename, date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last access date is to be modified. If you do not specify an absolute path, this function searches relative to the current working folder
date	Date variable containing the new last-accessed date

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetLastAccessDate function (it assumes an inv_filesrv instance variable):

```
Integer   li_return
String    ls_filename
Date      ldt_date

ls_filename = sle_source.text
ldt_date = Today()
li_return = inv_filesrv.of_SetLastAccessDate &
   (ls_filename, ldt_date)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
...
```

n_cst_filesrvsol2

Description

Platform-specific file management functions for applications running on UNIX under Solaris. Platform-specific file management functions require a call to a platform-specific external function. When you call the f_SetFilesrv global function, it detects the operating environment and instantiates the correct platform-specific object.

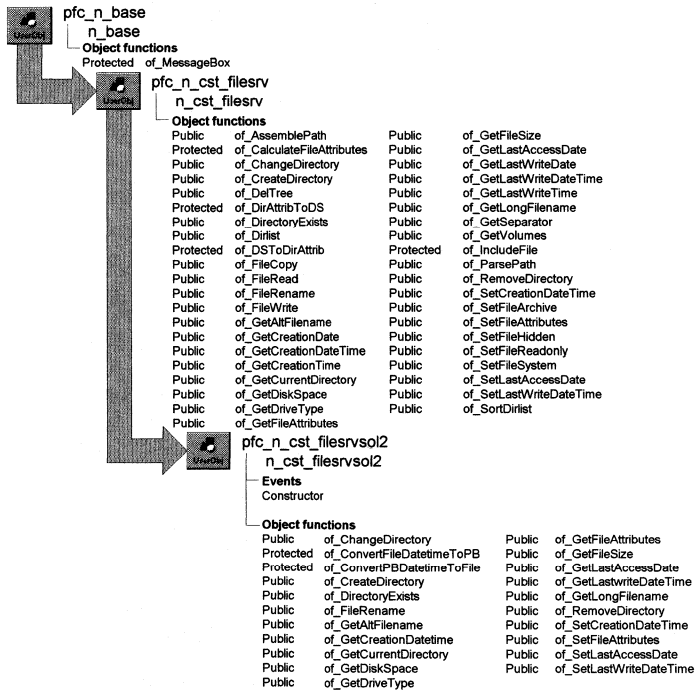
HP/UX and AIX

N_cst_filesrvaix and n_cst_filesrvhpux provide the identical functionality on IBM AIX and HP/UX.

Many of the functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar.

This object calls functions in the platform-specific module libkernal31.so.

Ancestry



Library

PFCAPSRV.PBL

PFEAPSRV.PBL

Object relationships

n_cst_dirattrib
n_cst_numerical

Usage

To use the file management service:

1 Instantiate the appropriate file-management object by calling the f_SetFilesrv global function. This function examines your environment and instantiates the appropriate user object.

2 Call file management functions as required. The code for these functions is defined in the descendent objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_filesrv) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_filedatetime	ul_lowdatetime	UnsignedLong
	ul_highdatetime	UnsignedLong
os_fileopeninfo	c_fixed_disk	Character
	c_length	Character
	c_pathname[128]	Character
	ui_dos_error	UnsignedInteger
	ui_na1	UnsignedInteger
	ui_na2	UnsignedInteger
os_finddata	ch_alternatefilename[14]	Character
	ch_filename[260]	Character
	str_creationtime	os_filedatetime
	str_lastaccesstime	os_filedatetime
	str_lastwritetime	os_filedatetime
	ul_fileattributes	UnsignedLong
	ul_filesizehigh	UnsignedLong
	ul_filesizelow	UnsignedLong
	ul_reserved0	UnsignedLong
	ul_reserved1	UnsignedLong

Structure	Field	Data type
os_systemtime	ui_wyear	Long
	ui_wmonth	Long
	ui_wdayofweek	Long
	ui_wday	Long
	ui_whour	Long
	ui_wminute	Long
	ui_wsecond	Long
	ui_wmilliseconds	Long
os_securityattributes	b_inherit	Boolean
	ch_description	Character
	ul_length	UnsignedLong

See also

f_SetFilesrv
n_cst_filesrv
n_cst_filesrvmac
n_cst_filesrvwin16
n_cst_filesrvwin32
n_cst_platform

Events

N_cst_filesrvsol2 contains a precoded event:

Constructor

Constructor

Description

Initializes the is_separator and is_allfiles instance variables.

Usage

This event executes when the object is created.

Functions

N_cst_filesrvsol2 includes precoded object functions:

of_ChangeDirectory	of_GetFileAttributes
of_ConvertFileDateTimeToPB	of_GetFileSize
of_ConvertPBDateTimeToFile	of_GetLastAccessDate
of_CreateDirectory	of_GetLastWriteDateTime
of_DirectoryExists	of_GetLongFilename
of_FileRename	of_RemoveDirectory
of_GetAltFilename	of_SetCreationDateTime
of_GetCreationDateTime	of_SetFileAttributes
of_GetCurrentDirectory	of_SetLastAccessDate
of_GetDiskSpace	of_SetLastWriteDateTime
of_GetDriveType	

of_ChangeDirectory

Description

Changes the current working directory.

Access

Public

Syntax

instancename.**of_ChangeDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the new working directory. If you do not specify an absolute path, this function changes directories relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to change the current folder. If your application performs functions that change the folder, call this function before path-sensitive PowerBuilder functions, such as ShowHelp, to ensure access to the proper files.

This function calls the SetCurrentDirectoryA external function.

Examples

This example changes folders to the apps/realapp directory before calling the SetProfileString function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_language

ls_language = sle_curr_language.Text
```

```
li_return = inv_filesrv.of_ChangeDirectory &
   ("apps/realapp")
IF li_return = 1 THEN
   SetProfileString("realapp.ini", "language", &
      "currentlanguage", ls_language)
END IF
```

of_ConvertFileDateTimeToPB

Description

Converts a file date (UnsignedLong) into a PowerBuilder date.

Access

Protected

Syntax

instancename.**of_ConvertFileDateTimeToPB** (*filedatetime*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filedatetime	Os_filedatetime object structure containing a file's system date and time
date	Date variable into which the function places the date (passed by reference)
time	Time variable into which the function places the time (passed by reference)

Return value

Date. Returns *date* and *time* in PowerBuilder format.

Usage

Internal.

This function calls the FileTimeToLocalFileTime and FileTimeToSystemTime external functions.

Examples

This example is from the of_DirList function:

```
Integer  li_cnt, li_entries
Long   ll_handle
String  ls_time, ls_date
Char  lc_drive
Boolean  lb_found
Time  lt_time
os_finddata  lstr_finddata
n_cst_dirattrib  lnv_empty[50]
n_cst_numerical  lnv_numeric
...
```

```
of_ConvertFileDateTimeToPB &
    (lstr_finddata.str_creationtime, &
      anv_dirlist[li_entries].id_creationdate, &
        anv_dirlist[li_entries].it_creationtime)
...
```

of_ConvertPBDateTimeToFile

Description

Converts a PowerBuilder date and time to a system date and time.

Access

Protected

Syntax

instancename.**of_ConvertPBDateTimeToFile** (*date*, *time*, *datetime*)

Argument	Description
instancename	Instance name of n_cst_filesrv
date	Date variable containing a date to be converted
time	Time variable containing a time to be converted
datetime	Os_filedatetime into which the function places the converted system date and time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

This function calls the SystemTimeToFileTime and LocalFileTimeToFileTime external functions.

Examples

This example is from the of_SetCreationDateTime function:

```
Boolean  lb_ret
Long     ll_error, ll_handle
os_filedatetime  lstr_filetime, lstr_empty
os_finddata    lstr_finddata
os_fileopeninfo  lstr_fileinfo
...
If of_ConvertPBDateTimeToFile (ad_date, at_Time, &
   lstr_filetime) < 0 THEN Return -1
...
```

of_CreateDirectory

Description: Creates a new directory.

Access: Public

Syntax: *instancename*.**of_CreateDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the new directory. If you do not specify an absolute path, this function creates a directory relative to the current working directory

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: This function calls the CreateDirectoryA external function.

Examples: This example calls the of_CreateDirectory function to create a temporary directory (it assumes an inv_filesrv instance variable):

```
IF inv_filesrv.of_CreateDirectory &
   ("temp") <> 0 THEN
      MessageBox("File Services", &
         "Error creating temp directory")
END IF
```

of_DirectoryExists

Description: Reports if the specified directory exists.

Access: Public

Syntax: *instancename*.**of_DirectoryExists** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the directory. If you do not specify an absolute path, this function looks for a directory relative to the current working directory

Return value: Boolean. Returns TRUE if the directory exists and FALSE if it does not.

Usage: This function calls the GetFileAttributesA external function.

Examples

This example calls the of_DirectoryExists function (it assumes an inv_filesrv instance variable):

```
Integer  li_return

IF NOT inv_filesrv.of_DirectoryExists &
   ("temp") THEN
      li_return = inv_filesrv.of_CreateDirectory &
         ("temp")
END IF
```

of_FileRename

Description

Renames or moves a file or folder.

Access

Public

Syntax

instancename.**of_FileRename** (*oldname*, *newname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
oldname	String specifying the complete path of the file to be renamed
newname	String specifying the complete path of the new filename

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the MoveFileA external function.

Examples

This example calls the of_FileRename function:

```
String   ls_source, ls_target
Integer  li_return

ls_source = sle_source.text
ls_target = sle_target.text
li_return = inv_filesrv.of_FileRename &
   (ls_source, ls_target)
IF li_return = 1 THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      (ls_source + " renamed to " + ls_target)
ELSE
   MessageBox("File Service", &
      "Error renaming " + ls_source + " to " &
```

```
            + ls_target)
END IF
```

of_GetAltFilename

Description — Retrieves the internal 8.3 alternate filename for a passed long filename.

Access — Public

Syntax — *instancename*.**of_GetAltFilename** (*longfilename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
longfilename	String specifying the long filename

Return value — String. Returns the alternate filename for *longfilename*.

Usage — This function calls the FindFirstFileA and FindClose external functions.

Examples — This example calls the of_GetAltFilename function (it assumes an inv_filesrv instance variable):

```
String  ls_filename, ls_return

ls_filename = sle_source.text
ls_return = inv_filesrv.of_GetAltFilename &
   (ls_filename)
IF ls_return = "" THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_return &
      + " is the alternate filename for " &
         + ls_filename)
END IF
```

of_GetCreationDateTime

Description — Retrieves the date and time a file was created.

Access — Public

Syntax — *instancename*.**of_GetCreationDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose creation date and time are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the creation date (passed by reference)
time	Time into which the function places the creation time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetCreationDateTime function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetCreationDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + "~r~nCreation date: " &
      + String(ldt_date) &
      + "~r~nCreation time: " &
      + String(lt_time) )
END IF
```

of_GetCurrentDirectory

Description

Retrieves the current working directory.

Access

Public

Syntax

instancename.**of_GetCurrentDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_filesrv

Return value

String. Returns the fully qualified path of the current working directory.

Usage

This function calls the GetCurrentDirectoryA external function.

Examples

This example calls the of_GetCurrentDirectory function and, if the result isn't equal to the is_app_dir instance variable, calls the of_ChangeDirectory function to reset the working directory (it assumes an inv_filesrv instance variable):

```
String  ls_return

ls_return = inv_filesrv.of_GetCurrentDirectory()
IF ls_return <> is_app_dir THEN
   inv_filesrv.of_ChangeDirectory(is_app_dir)
END IF
```

of_GetDiskSpace

Description

Retrieves space information for a specified disk drive.

Access

Public

Syntax

instancename.**of_GetDiskSpace** (*drive, total, free*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter
total	Long into which the function returns the total bytes on *drive* (passed by reference)
free	Long into which the function returns the free space on *drive* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetDiskFreeSpaceA external function.

Examples

This example calls the of_GetDiskSpace function:

```
Character  lc_drive
```

```
Long        ll_total, ll_free
Integer     li_return

lc_drive = lb_drive.SelectedItem()
li_return = inv_filesrv.of_GetDiskSpace &
   (lc_drive, ll_total, ll_free)
IF li_return = 1 THEN
   MessageBox("File Service", "Drive " + lc_drive &
   + "~r~n" + String(ll_total) + " total bytes~r~n" &
   + String(ll_free) + " total free space")
ELSE
   MessageBox("File Service", &
      "Error in of_GetDiskSpace")
END IF
```

of_GetDriveType

Description

Determines the drive type for a specified drive.

Access

Public

Syntax

instancename.**of_GetDriveType** (*drive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter

Return value

Integer. Returns values as follows:

0 Undetermined drive type
1 Root directory does not exist
2 Removable disk
3 Nonremovable disk
4 Remote (network) drive
5 CD-ROM drive
6 RAM disk

Usage

This function calls the GetDriveTypeA external function.

Examples

This example calls the of_GetDriveType function (it assumes an inv_filesrv instance variable):

```
Integer  lui_drivetype
String   ls_driveletter
```

```
String   ls_drivetype[ ] = &
{"Undetermined","Does not exist","Removeable", &
   "Nonremoveable", "Network drive", "CD-ROM", &
      "RAM disk" }

ls_driveletter = lb_drivelist.SelectedItem()
lui_drivetype = inv_filesrv.of_GetDriveType &
   (ls_driveletter)
MessageBox("File Service", "Drive type is: " &
   + ls_drivetype[lui_drivetype + 1] )
```

of_GetFileAttributes

Description

Accesses all attributes for a specified file.

Access

Public

Syntax

instancename.**of_GetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *subfolder*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose attributes are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
readonly	Boolean indicating whether *filename* is read-only (passed by reference)
hidden	Boolean indicating whether *filename* is hidden (passed by reference)
system	Boolean indicating whether *filename* is a system file (passed by reference)
subfolder	Boolean indicating whether *filename* is a subfolder (passed by reference)
archive	Boolean indicating whether *filename* has the archive bit set (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetFileAttributes function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
     cbx_hidden.Checked, &
     cbx_system.Checked, &
     cbx_subdirectory.Checked, &
     cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
     "Error accessing " + ls_filename)
END IF
```

of_GetFileSize

Description

Retrieves the size of a specified file.

Access

Public

Syntax

instancename.**of_GetFileSize** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose size is to be returned. If you do not specify an absolute path, this function searches relative to the current working directory

Return value

Double. Returns the number of bytes in *filename* if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetFileSize function (it assumes an inv_filesrv instance variable):

```
Double   ld_return
String   ls_filename

ls_filename = sle_source.text
```

```
ld_return = &
   inv_filesrv.of_GetFileSize(ls_filename)
IF ld_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a size of " + String(ld_return) &
      + " bytes.")
END IF
```

of_GetLastAccessDate

Description

Determines the date a file was last accessed.

Access

Public

Syntax

instancename.**of_GetLastAccessDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last access date is to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the last access date (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetLastAccessDate function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetLastAccessDate &
   (ls_filename, ldt_date)
IF li_return = -1 THEN
   MessageBox("File Service", &
```

```
        "Error accessing " + ls_filename)
ELSE
    MessageBox("File Service", ls_filename &
        + " has a last access date of " &
        + String(ldt_date) )
END IF
```

of_GetLastWriteDateTime

Description

Determines the date and time a file was last modified.

Access

Public

Syntax

instancename.**of_GetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last write date and time are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the last write date (passed by reference)
time	Time into which the function places the last write time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetLastWriteDateTime function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetLastWriteDateTime &
    (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
    MessageBox("File Service", &
        "Error accessing " + ls_filename)
```

```
ELSE
   MessageBox("File Service", ls_filename &
      + " has a last write date of " &
      + String(ldt_date) &
      + " and a last write time of &
      " + String(lt_time) )
END IF
```

of_GetLongFilename

Description

Retrieves the corresponding long filename for a passed internal 8.3 (also called alternate) filename.

Access

Public

Syntax

instancename.**of_GetLongFilename** (*shortfilename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
shortfilename	String specifying the short (alternate) filename

Return value

String. Returns the long filename for *shortfilename*.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetLongFilename function (it assumes an inv_filesrv instance variable):

```
String  ls_filename, ls_return

ls_filename = sle_source.text
ls_return = inv_filesrv.of_GetLongFilename &
   (ls_filename)
IF ls_return = "" THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_return &
      + " is the long filename for " &
         + ls_filename)
END IF
```

of_RemoveDirectory

Description

Deletes the specified directory (the directory must be empty). To delete a directory and its files, call of_DelTree instead.

Access

Public

Syntax

instancename.**of_RemoveDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrvsol2
directoryname	String specifying the path to the directory to be deleted. If you do not specify an absolute path, this function deletes relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the RemoveDirectoryA external function.

Examples

This example checks if pbapp is the current directory and, if it is, deletes the pbapp/temp directory (it assumes an inv_filesrv instance variable):

```
String   ls_return, ls_temp
Integer  li_return

ls_temp = "temp"
ls_return = &
gnv_app.inv_platform.of_GetCurrentDirectory( )
IF Mid(ls_return, 4, 5) = "pbapp" THEN
   li_return = inv_filesrv.of_RemoveDirectory &
      (ls_temp)
   IF li_return = 1 THEN
      MessageBox("Delete", &
         "Temp folder does not exist")
   ELSEIF li_return <0 THEN
      MessageBox("Delete", &
         "Deletion error: Temp dir must be empty")
   END IF
ELSE
   MessageBox("Delete", &
      "Cannot delete temp. Change dir to pbapp")
   Return
END IF
```

of_SetCreationDateTime

Description

Sets a file's creation date and time.

Access

Public

Syntax

instancename.**of_SetCreationDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose creation date and time are to be modified. If you do not specify an absolute path, this function searches relative to the current working folder
date	Date variable containing the new creation date
time	Time variable containing the new creation time

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA, FindClose, OpenFile, SetFileTime, and CloseHandle external functions.

Examples

This example calls the of_SetCreationDateTime function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesrv.of_SetCreationDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a new creation date of " &
      + String(ldt_date) &
      + " and a new creation time of " &
      + String(lt_time) )
```

```
END IF
```

of_SetFileAttributes

Description

Sets attributes for the specified file.

Access

Public

Syntax

instancename.**of_SetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrvsol2
filename	String specifying the file whose attributes are to be set. If you do not specify an absolute path, this function searches relative to the current working directory
readonly	Boolean indicating whether *filename* is read-only. If you specify NULL, the function doesn't modify this attribute
hidden	Boolean indicating whether *filename* is hidden. If you specify NULL, the function doesn't modify this attribute
system	Boolean indicating whether *filename* is a system file. If you specify NULL, the function doesn't modify this attribute
archive	Boolean indicating whether *filename* has the archive bit set. If you specify NULL, the function doesn't modify this attribute

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the SetFileAttributesA external function.

Examples

This example calls the of_SetFileAttributes function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_SetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
```

```
        MessageBox("File Service", &
            "Error accessing " + ls_filename)
    END IF
```

of_SetLastAccessDate

Description — Sets a file's last-accessed date and time.

Access — Public

Syntax — *instancename*.**of_SetLastAccessDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last access date is to be modified. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date variable containing the new last-accessed date

Return value — Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage — This function calls the OpenFile and CloseHandle external functions.

Examples — This example calls the of_SetLastAccessDate function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date

ls_filename = sle_source.text
ldt_date = Today()
li_return = inv_filesrv.of_SetLastAccessDate &
   (ls_filename, ldt_date)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a new last access date of " &
         + String(ldt_date))
END IF
```

of_SetLastWriteDateTime

Description

Sets a file's last-modified date and time.

Access

Public

Syntax

instancename.**of_SetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last write date and time are to be modified. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date variable containing the new last-modified date
time	Time variable containing the new last-modified time

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the FindFirstFileA, FindClose, OpenFile, SetFileTime, and CloseHandle external functions.

Examples

This example calls the of_SetLastWriteDateTime function (it assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesrv.of_SetLastWriteDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " New last write date = " String(ldt_date) &
      + " New last write time = " String(lt_time))
END IF
```

n_cst_filesrvwin16

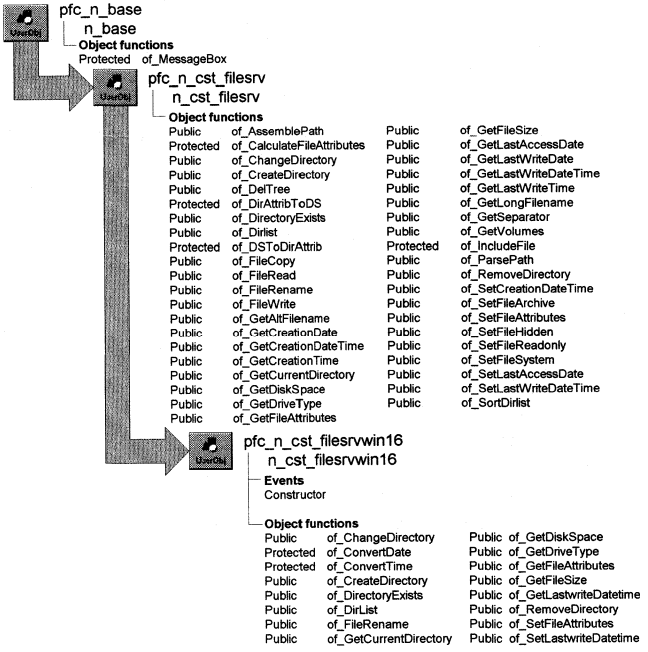
Description

Platform-specific file management functions for applications running under Windows 3.1. Platform-specific file management functions require a call to a platform-specific external function. When you call the f_SetFilesrv global function, it detects the operating environment and instantiates the correct platform-specific object.

Many of the functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar. This object calls function in the following platform-specific modules:

PFCFLSRV.DLL, shipped with PFC
KERNEL.EXE, shipped with Windows 3.1

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_dirattrib
n_cst_numerical

Usage

To use the file management service:

1 Instantiate the appropriate file management object by calling the f_SetFilesrv global function. This function examines your environment and instantiates the appropriate user object.

2 Call file management functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_filesrv) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_diskfreet	ui_availclusters	UnsignedInteger
	ui_bytespersector	UnsignedInteger
	ui_sectorspercluster	UnsignedInteger
	ui_totalclusters	UnsignedInteger
os_findt	ch_attrib	Character
	ch_name[13]	Character
	ch_reserved[21]	Character
	ui_wrdate	UnsignedInteger
	ui_wrtime	UnsignedInteger
	ul_size	UnsignedLong

See also

f_SetFilesrv
n_cst_filesrv
n_cst_filesrvmac
n_cst_filesrvsol2
n_cst_filesrvwin32
n_cst_platform

Events

N_cst_filesrvwin16 contains a precoded event:

Constructor

Constructor

Description: Initializes the is_separator and is_allfiles instance variables.

Usage: This event executes when the object is created.

Functions

N_cst_filesrvwin16 includes precoded object functions:

of_ChangeDirectory	of_GetDiskSpace
of_ConvertDate	of_GetDriveType
of_ConvertTime	of_GetFileAttributes
of_CreateDirectory	of_GetFileSize
of_DirectoryExists	of_GetLastWriteDateTime
of_DirList	of_RemoveDirectory
of_FileRename	of_SetFileAttributes
of_GetCurrentDirectory	of_SetLastWriteDateTime

of_ChangeDirectory

Description: Changes the current working directory.

Access: Public

Syntax: *instancename*.**of_ChangeDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the new working directory. If you do not specify an absolute path, this function changes directories relative to the current working directory

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: If your application performs functions that change the working directory, call this function before path-sensitive PowerBuilder functions, such as ShowHelp, to ensure access to the proper files. This enables your applications to work without depending on the system path.

This function calls the ChangeDirectory external function.

Examples

This example changes directory to the \realapp directory before calling the SetProfileString function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_language

ls_language = sle_curr_language.Text
li_return = inv_filesrv.of_ChangeDirectory &
   ("\realapp")
IF li_return = 1 THEN
   SetProfileString("realapp.ini", "language", &
      "currentlanguage", ls_language)
END IF
```

of_ConvertDate

Description

Converts a file date (UnsignedInteger) into a PowerBuilder date.

Access

Protected

Syntax

instancename.**of_ConvertDate** (*filedate*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filedate	UnsignedInteger containing the file date

Return value

Date. Returns *filedate* in PowerBuilder format.

Usage

Internal.

This function calls the ConvDate external function.

Examples

This example is from the of_DirList function:

```
Integer  li_Cnt, li_Entries, li_RC
Uint     lui_Mask, lui_Attrib
String   ls_Time, ls_Date
Char     lc_Drive
os_findt  lstr_FindData
n_cst_dirattrib  lnv_Empty[ ]
n_cst_numerical  lnv_Numeric
...
IF of_IncludeFile(String(lstr_FindData.c_Name), &
```

```
        al_FileType, lui_Attrib) THEN
          li_Entries ++
          anv_DirList[li_Entries].is_FileName = &
            lstr_FindData.c_Name
          anv_DirList[li_Entries].id_LastWriteDate = &
            of_ConvertDate(lstr_FindData.ui_wr_Date)
          anv_DirList[li_Entries].it_LastWriteTime = &
            of_ConvertTime(lstr_FindData.ui_wr_Time)
...
```

of_ConvertTime

Description Converts a file time (UnsignedInteger) into a PowerBuilder time.

Access Protected

Syntax *instancename*.**of_ConvertTime** (*filetime*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filetime	UnsignedInteger containing the file time

Return value Time. Returns *filetime* in PowerBuilder format.

Usage Internal.

This function calls the ConvTime external function.

Examples This example is from the of_DirList function:

```
...
anv_DirList[li_Entries].it_LastWriteTime = &
   of_ConvertTime(lstr_FindData.ui_wr_Time)
...
```

of_CreateDirectory

Description Creates a new directory.

Access Public

Syntax *instancename*.**of_CreateDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the new directory. If you do not specify an absolute path, this function creates a directory relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the DirectoryCreate external function.

Examples

This example calls the of_CreateDirectory function to create a temporary directory (this example assumes an inv_filesrv instance variable):

```
IF inv_filesrv.of_CreateDirectory &
   ("temp") <> 0 THEN
      MessageBox("File Services", &
         "Error creating temp directory")
END IF
```

of_DirectoryExists

Description

Reports whether the specified directory exists.

Access

Public

Syntax

instancename.**of_DirectoryExists** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the directory. If you do not specify an absolute path, this function looks for a directory relative to the current working directory

Return value

Boolean. Returns TRUE if the directory exists and FALSE if it does not.

Usage

This function calls the DirectoryExists external function.

Examples

This example calls the of_DirectoryExists function (this example assumes an inv_filesrv instance variable):

```
Integer   li_return

IF NOT inv_filesrv.of_DirectoryExists &
```

```
("c:\rkndoc\temp") THEN
   li_return = inv_filesrv.of_CreateDirectory &
      ("c:\rkndoc\temp")
END IF
```

of_DirList

Description

Retrieves information on all files within the passed directory that match the passed arguments.

Access

Public

Syntax

instancename.**of_DirList** (*filespec*, *filetype*, *dirlist*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filespec	String specifying the path and file specification (including wildcards) to the directory containing the files to be listed. If you do not specify an absolute path, this function searches relative to the current working directory
filetype	Long specifying one or more types of files to return. Types are: ♦ 0—Read/write files ♦ 1—Read-only files ♦ 2—Hidden files ♦ 4—System files ♦ 16—Subdirectories ♦ 32—Archive (modified) files ♦ 16384—Drives ♦ 32768—Exclude read/write files from the list
dirlist	Array of type n_cst_dirattrib into which the function places information for the requested files (passed by reference)

Return value

Integer. Returns the number of elements in the *dirlist* array if the function succeeds and -1 if an error occurs.

Usage

To further sort the *dirlist* array, use the of_SortDirList function.

To list several types, add the *filetype* values together. For example, to list read/write files, subdirectories, and drives, specify 0+16+16384 or 16400 for *filetype*.

Examples

This example calls the of_DirList function (it assumes an inv_filesrv instance variable):

```
String   ls_currdir
Integer  li_cnt, li_entries
String   ls_import
n_cst_dirattrib  lnv_dirlist[ ]

SetPointer(HourGlass!)
ls_currdir = "c:\Win95\*.dll"

li_entries = &
   inv_filesrv.of_DirList &
      (ls_currdir, 0, lnv_dirlist)
...
```

of_FileRename

Description

Renames or moves a file or directory.

Access

Public

Syntax

instancename.**of_FileRename** (*oldname*, *newname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
oldname	String specifying the complete path of the file to be renamed
newname	String specifying the complete path of the new filename

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the RenameFile external function.

Examples

This example calls the of_FileRename function:

```
String   ls_source, ls_target
Integer  li_return

ls_source = sle_source.text
ls_target = sle_target.text
li_return = inv_filesrv.of_FileRename &
   (ls_source, ls_target)
IF li_return = 1 THEN
```

```
        gnv_app.of_GetFrame().SetMicroHelp &
            (ls_source + " renamed to " + ls_target)
    ELSE
        MessageBox("File Service", &
            "Error renaming " + ls_source + " to " &
                + ls_target)
    END IF
```

of_GetCurrentDirectory

Description — Retrieves the current working directory.

Access — Public

Syntax — *instancename*.**of_GetCurrentDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_filesrv

Return value — String. Returns the fully qualified path of the current working directory.

Usage — This function calls the CurrentDirectory external function.

Examples — This example calls the of_GetCurrentDirectory function (it assumes an inv_filesrv instance variable):

```
String   ls_return

ls_return = inv_filesrv.of_GetCurrentDirectory()
IF ls_return <> is_app_dir THEN
    inv_filesrv.of_ChangeDirectory(is_app_dir)
END IF
```

of_GetDiskSpace

Description — Retrieves space information for a specified disk drive.

Access — Public

Syntax — *instancename*.**of_GetDiskSpace** (*drive*, *total*, *free*)

Argument	Description
instancename	Instance name of n_cst_filesrv

Argument	Description
drive	Character specifying the drive letter
total	Long into which the function returns the total bytes on *drive* (passed by reference)
free	Long into which the function returns the free space on *drive* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetDiskSpace external function.

Examples

This example calls the of_GetDiskSpace function:

```
Character  lc_drive
Long   ll_total, ll_free
Integer   li_return

lc_drive = lb_drive.SelectedItem()
li_return = inv_filesrv.of_GetDiskSpace &
   (lc_drive, ll_total, ll_free)
IF li_return = 1 THEN
   MessageBox("File Service", "Drive " + lc_drive &
   + "~r~n" + String(ll_total) + "total bytes~r~n" &
      + String(ll_free) + " total free space")
ELSE
   MessageBox("File Service", &
      "Error in of_GetDiskSpace")
END IF
```

of_GetDriveType

Description

Determines the drive type for a specified drive.

Access

Public

Syntax

instancename.**of_GetDriveType** (*drive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter

Return value

Integer. Returns values as follows:

0 Undetermined drive type
1 Root directory does not exist
2 Removable disk
3 Nonremovable disk
4 Remote (network) drive
5 CD-ROM drive
6 RAM disk

Examples

This example calls the of_GetDriveType function (this example assumes an inv_filesrv instance variable):

```
UnsignedInteger  lui_drivetype
String  ls_driveletter
String  ls_drivetype[ ] = &
   {"Undetermined","Does not exist","Removeable", &
     "Nonremoveable", "Network drive", "CD-ROM", &
       "RAM disk" }

ls_driveletter = lb_drivelist.SelectedItem( )
lui_drivetype = inv_filesrv.of_GetDriveType &
   (ls_driveletter)
MessageBox("File Service", "Drive type is: " &
   + ls_drivetype[lui_drivetype + 1] )
```

of_GetFileAttributes

Description

Retrieves all attributes for a specified file.

Access

Public

Syntax

instancename.**of_GetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *subdirectory*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose attributes are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
readonly	Boolean indicating whether *filename* is read-only (passed by reference)
hidden	Boolean indicating whether *filename* is hidden (passed by reference)

Argument	Description
system	Boolean indicating whether *filename* is a system file (passed by reference)
subdirectory	Boolean indicating whether *filename* is a subdirectory (passed by reference)
archive	Boolean indicating whether *filename* has the archive bit set (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetFileAttributes function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_subdirectory.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
END IF
```

of_GetFileSize

Description

Retrieves the size of a specified file.

Access

Public

Syntax

instancename.**of_GetFileSize** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose size is to be returned. If you do not specify an absolute path, this function searches relative to the current working directory

Return value

Double. Returns the number of bytes in *filename* if the function succeeds and -1 if an error occurs.

Usage

This function calls the Findfirst and FindClose external functions.

Examples

This example calls the of_GetFileSize function (this example assumes an inv_filesrv instance variable):

```
Double   ld_return
String   ls_filename

ls_filename = sle_source.text
ld_return = &
   inv_filesrv.of_GetFileSize(ls_filename)
IF ld_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a size of " + String(ld_return) &
         + " bytes.")
END IF
```

of_GetLastWriteDateTime

Description

Retrieves the date and time a file was last modified.

Access

Public

Syntax

instancename.**of_GetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last write date and time are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the last write date (passed by reference)
time	Time into which the function places the last write time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetLastWriteDateTime function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetLastWriteDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a last write date of " &
      + String(ldt_date) &
      + " and a last write time of " &
      + String(lt_time) )
END IF
```

of_RemoveDirectory

Description

Deletes the specified directory (the directory must be empty). To delete a directory and its files, call of_DelTree instead.

Access

Public

Syntax

instancename.**of_RemoveDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrvwin16
directoryname	String specifying the path to the directory to be deleted. If you do not specify an absolute path, this function deletes relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the DirectoryRemove external function.

Examples

This example calls the of_RemoveDirectory function (this example assumes an inv_filesrv instance variable):

```
String   ls_return, ls_temp
Integer  li_return

ls_temp = "temp"
ls_return =
   gnv_app.inv_platform.of_GetCurrentDirectory( )
IF Mid(ls_return, 4, 5) = "pbapp" THEN
   li_return = inv_filesrv.of_RemoveDirectory &
      (ls_temp)
   IF li_return = 1 THEN
      MessageBox("Delete", &
         "Temp dir does not exist")
   ELSEIF li_return <0 THEN
      MessageBox("Delete", &
         "Deletion error: Temp dir must be empty")
   END IF
...
```

of_SetFileAttributes

Description

Sets attributes for the specified file.

Access

Public

Syntax

instancename.**of_SetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *subdirectory*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrvwin16
filename	String specifying the file whose attributes are to be set. If you do not specify an absolute path, this function searches relative to the current working directory
readonly	Boolean indicating whether *filename* is read-only. If you specify NULL, the function doesn't modify this attribute
hidden	Boolean indicating whether *filename* is hidden. If you specify NULL, the function doesn't modify this attribute
system	Boolean indicating whether *filename* is a system file. If you specify NULL, the function doesn't modify this attribute
subdirectory	Boolean indicating whether *filename* is a subdirectory. If you specify NULL, the function doesn't modify this attribute

Argument	Description
archive	Boolean indicating whether *filename* has the archive bit set. If you specify NULL, the function doesn't modify this attribute

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileAttributes function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_SetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
END IF
```

of_SetLastWriteDateTime

Description

Sets a file's last-modified date and time.

Access

Public

Syntax

instancename.**of_SetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last write date and time are to be modified. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date variable containing the new last-modified date
time	Time variable containing the new last-modified time

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetLastWriteDateTime function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesrv.of_SetLastWriteDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a new last write date of " &
      + String(ldt_date) &
      + " and a new last write time of " &
      + String(lt_time) )
END IF
```

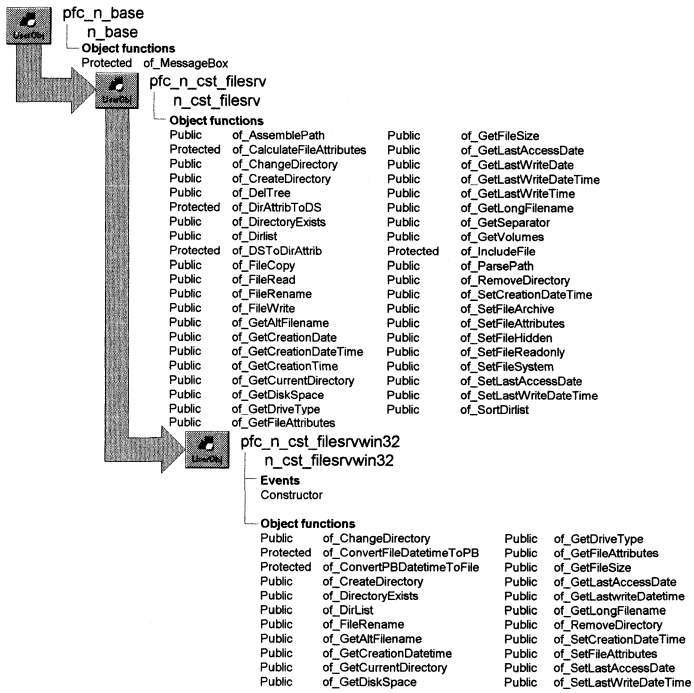
n_cst_filesrvwin32

Description

Platform-specific file management functions for applications running under Windows 95 and Windows NT. Platform-specific file management functions require a call to a platform-specific external function. When you call the f_SetFilesrv global function, it detects the operating environment and instantiates the correct platform-specific object.

Many of the functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar. This object calls functions in the KERNEL32.DLL module.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_dirattrib
n_cst_numerical

Usage

To use the file management service:

1 Instantiate the appropriate file management object by calling the f_SetFilesrv global function. This function examines your environment and instantiates the appropriate user object.

2 Call file management functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_filesrv) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_filedatetime	ul_highdatetime	UnsignedLong
	ul_lowdatetime	UnsignedLong
os_systemtime	ui_wday	UnsignedInteger
	ui_wdayofweek	UnsignedInteger
	ui_whour	UnsignedInteger
	ui_wmilliseconds	UnsignedInteger
	ui_wminute	UnsignedInteger
	ui_wmonth	UnsignedInteger
	ui_wsecond	UnsignedInteger
	ui_wyear	UnsignedInteger
os_securityattributes	b_inherit	Boolean
	ch_description	Character
	ul_length	UnsignedLong
os_finddata	ch_alternatefilename[14]	Character
	ch_filename[260]	Character
	str_creationtime	os_filedatetime
	str_lastaccesstime	os_filedatetime
	str_lastwritetime	os_filedatetime
	ul_fileattributes	UnsignedLong
	ul_filesizehigh	UnsignedLong
	ul_filesizelow	UnsignedLong

Structure	Field	Data type
	ul_reserved0	UnsignedLong
	ul_reserved1	UnsignedLong
os_fileopeninfo	c_fixed_disk	Character
	c_length	Character
	c_pathname[128]	Character
	ui_dos_error	UnsignedInteger
	ui_na1	UnsignedInteger
	ui_na2	UnsignedInteger

See also

f_SetFilesrv
n_cst_filesrv
n_cst_filesrvmac
n_cst_filesrvsol2
n_cst_filesrvwin16
n_cst_platform

Events

N_cst_filesrvwin32 contains a precoded event:

Constructor

Constructor

Description

Initializes the is_separator and is_allfiles instance variables.

Usage

This event executes when the object is created.

Functions

N_cst_filesrvwin32 includes precoded object functions:

of_ChangeDirectory | of_GetDriveType

of_ConvertFileDateTimeToPB	of_GetFileAttributes
of_ConvertPBDateTimeToFile	of_GetFileSize
of_CreateDirectory	of_GetLastAccessDate
of_DirectoryExists	of_GetLastWriteDateTime
of_DirList	of_GetLongFilename
of_FileRename	of_RemoveDirectory
of_GetAltFilename	of_SetCreationDateTime
of_GetCreationDateTime	of_SetFileAttributes
of_GetCurrentDirectory	of_SetLastAccessDate
of_GetDiskSpace	of_SetLastWriteDateTime

of_ChangeDirectory

Description

Changes the current working directory.

Access

Public

Syntax

instancename.**of_ChangeDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the new working directory. If you do not specify an absolute path, this function changes directories relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If your application performs functions that change the working directory, call this function before path-sensitive PowerBuilder functions, such as ShowHelp, to ensure access to the proper files. This enables your applications to work without depending on the system path.

This function calls the SetCurrentDirectory external function.

Examples

This example changes directory to the \realapp directory before calling the SetProfileString function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String  ls_language

ls_language = sle_curr_language.Text
li_return = inv_filesrv.of_ChangeDirectory &
    ("\realapp")
IF li_return = 1 THEN
```

```
        SetProfileString("realapp.ini", "language", &
           "currentlanguage", ls_language)
    END IF
```

of_ConvertFileDateTimeToPB

Description

Converts a file date (UnsignedInteger) into a PowerBuilder date.

Access

Protected

Syntax

instancename.**of_ConvertFileDateTimeToPB** (*filedatetime*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filedatetime	Structure of type os_filedatetime containing a file's system date and time
date	Date variable into which the function places the date (passed by reference)
time	Time variable into which the function places the time (passed by reference)

Return value

Date. Returns *date* and *time* in PowerBuilder format.

Usage

Internal.

This function calls the FileTimeToLocalFileTime and FileTimeToSystemTime external functions.

Examples

This example is from the of_DirList function:

```
os_finddata   lstr_FindData
...
of_ConvertFileDateTimeToPB &
    (lstr_FindData.str_CreationTime, &
       anv_DirList[li_Entries].id_CreationDate, &
       anv_DirList[li_Entries].it_CreationTime)
...
```

of_ConvertPBDateTimeToFile

Description

Converts a PowerBuilder date and time to a system date and time.

Access

Protected

Syntax

instancename.**of_ConvertPBDateTimeToFile** (*date*, *time*, *datetime*)

Argument	Description
instancename	Instance name of n_cst_filesrv
date	Date variable containing a date to be converted
time	Time variable containing a time to be converted
datetime	Structure of type of_filedatetime into which the function places the converted system date and time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

This function calls the SystemTimeToFileTime and LocalFileTimeToFileTime external functions.

Examples

This example is from the of_SetLastWriteDateTime function:

```
Boolean  lb_Ret
Long   ll_Error, ll_Handle
os_filedatetime  lstr_FileTime, lstr_Empty
os_finddata   lstr_FindData
os_fileopeninfo   lstr_FileInfo

ll_Handle = FindFirstFileA(as_FileName, lstr_FindData)
IF ll_Handle <= 0 THEN Return -1
   FindClose(ll_Handle)

IF of_ConvertPBDatetimeToFile &
   (ad_Date, at_Time, lstr_FileTime) < 0 THEN
      Return 1
END IF
...
```

of_CreateDirectory

Description

Creates a new directory.

Access

Public

Syntax

instancename.**of_CreateDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the new directory. If you do not specify an absolute path, this function creates a directory relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the CreateDirectoryA external function.

Examples

This example calls the of_CreateDirectory function (this example assumes an inv_filesrv instance variable):

```
IF inv_filesrv.of_CreateDirectory &
   ("temp") <> 0 THEN
      MessageBox("File Services", &
        "Error creating temp directory")
END IF
```

of_DirectoryExists

Description

Reports whether the specified directory exists.

Access

Public

Syntax

instancename.**of_DirectoryExists** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
directoryname	String specifying the name of the directory. If you do not specify an absolute path, this function looks for a directory relative to the current working directory

Return value

Boolean. Returns TRUE if the directory exists and FALSE if it does not.

Usage

This function calls the GetFileAttributesA external function.

Examples

This example calls the of_DirectoryExists function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return

IF NOT inv_filesrv.of_DirectoryExists &
```

```
    ("c:\rkndoc\temp") THEN
      li_return = inv_filesrv.of_CreateDirectory &
        ("c:\rkndoc\temp")
END IF
```

of_DirList

Description

Retrieves information on all files within the passed directory that match the passed arguments.

Access

Public

Syntax

instancename.**of_DirList** (*filespec*, *filetype*, *dirlist*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filespec	String specifying the path and file specification (including wildcards) to the directory containing the files to be listed. If you do not specify an absolute path, this function searches relative to the current working directory
filetype	Long specifying one or more types of files to return. Types are: ♦ 0—Read/write files ♦ 1—Read-only files ♦ 2—Hidden files ♦ 4—System files ♦ 16—Subdirectories ♦ 32—Archive (modified) files ♦ 16384—Drives ♦ 32768—Exclude read/write files from the list
dirlist	Array of type n_cst_dirattrib into which the function places information for the requested files (passed by reference)

Return value

Integer. Returns the number of elements in the *dirlist* array if the function succeeds and -1 if an error occurs.

Usage

To further sort the *dirlist* array, use the of_SortDirList function.

To list several types, add the *filetype* values together. For example, to list read/write files, subdirectories, and drives, specify 0+16+16384 or 16400 for *filetype*.

This function calls the FindNextFileA external function.

Examples

This example calls the of_DirList function (this example assumes an inv_filesrv instance variable):

```
String   ls_currdir
Integer  li_cnt, li_entries
String   ls_import
n_cst_dirattrib   lnv_dirlist[]

SetPointer(HourGlass!)
ls_currdir = "c:\Win95\*.dll"

li_entries = &
   inv_filesrv.of_DirList &
      (ls_currdir, 0, lnv_dirlist)
...
```

of_FileRename

Description

Renames or moves a file or directory.

Access

Public

Syntax

instancename.**of_FileRename** (*oldname*, *newname*)

Argument	Description
instancename	Instance name of n_cst_filesrv
oldname	String specifying the complete path of the file to be renamed
newname	String specifying the complete path of the new filename

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the MoveFileA external function.

Examples

This example calls the of_FileRename function:

```
String   ls_source, ls_target
Integer  li_return

ls_source = sle_source.text
ls_target = sle_target.text
li_return = inv_filesrv.of_FileRename &
```

```
    (ls_source, ls_target)
IF li_return = 1 THEN
    gnv_app.of_GetFrame().SetMicroHelp &
        (ls_source + " renamed to " + ls_target)
ELSE
    MessageBox("File Service", &
        "Error renaming " + ls_source + " to " &
            + ls_target)
END IF
```

of_GetAltFilename

Description

Retrieves the internal 8.3 alternate filename for a passed long filename.

Access

Public

Syntax

instancename.**of_GetAltFilename** (*longfilename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
longfilename	String specifying the long filename

Return value

String. Returns the alternate filename for *longfilename*.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetAltFilename function (this example assumes an inv_filesrv instance variable):

```
String   ls_filename, ls_return

ls_filename = sle_source.text
ls_return = inv_filesrv.of_GetAltFilename &
    (ls_filename)
IF ls_return = "" THEN
    MessageBox("File Service", &
        "Error accessing " + ls_filename)
ELSE
    MessageBox("File Service", ls_return &
        + " is the alternate filename for " &
        + ls_filename)
END IF
```

of_GetCreationDateTime

Description

Retrieves the date and time a file was created.

Access

Public

Syntax

instancename.**of_GetCreationDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose creation date and time are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the creation date (passed by reference)
time	Time into which the function places the creation time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetCreationDateTime function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date     ldt_date
Time     lt_time

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetCreationDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + "~r~nCreation date: " &
      + String(ldt_date) &
      + "~r~nCreation time: " &
      + String(lt_time) )
END IF
```

of_GetCurrentDirectory

Description: Retrieves the current working directory.

Access: Public

Syntax: *instancename*.**of_GetCurrentDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_filesrv

Return value: String. Returns the fully qualified path of the current working directory.

Usage: This function calls the GetCurrentDirectoryA external function.

Examples: This example calls the of_GetCurrentDirectory function and, if the result isn't equal to the is_app_dir instance variable, calls the of_ChangeDirectory function to reset the working directory (this example assumes an inv_filesrv instance variable):

```
String   ls_return

ls_return = inv_filesrv.of_GetCurrentDirectory()
IF ls_return <> is_app_dir THEN
   inv_filesrv.of_ChangeDirectory(is_app_dir)
END IF
```

of_GetDiskSpace

Description: Retrieves space information for a specified disk drive.

Access: Public

Syntax: *instancename*.**of_GetDiskSpace** (*drive*, *total*, *free*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter
total	Long into which the function returns the total bytes on *drive* (passed by reference)
free	Long into which the function returns the free space on *drive* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the GetDiskFreeSpaceA external function.

Examples

This example calls the of_GetDiskSpace function:

```
Character   lc_drive
Long   ll_total, ll_free
Integer   li_return

lc_drive = lb_drive.SelectedItem()
li_return = inv_filesrv.of_GetDiskSpace &
   (lc_drive, ll_total, ll_free)
IF li_return = 1 THEN
   MessageBox("File Service", "Drive " + lc_drive &
   + "~r~n" + String(ll_total) + " total bytes~r~n" &
      + String(ll_free) + " total free space")
ELSE
   MessageBox("File Service", &
      "Error in of_GetDiskSpace")
END IF
```

of_GetDriveType

Description

Retrieves the drive type for a specified drive.

Access

Public

Syntax

instancename.**of_GetDriveType** (*drive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
drive	Character specifying the drive letter

Return value

Integer. Returns values as follows:

0 Undetermined drive type
1 Root directory does not exist
2 Removable disk
3 Nonremovable disk
4 Remote (network) drive
5 CD-ROM drive
6 RAM disk

Examples

This example calls the of_GetDriveType function (this example assumes an inv_filesrv instance variable):

```
UnsignedInteger  lui_drivetype
String   ls_driveletter
String    ls_drivetype[ ] = &
   {"Undetermined","Does not exist","Removeable", &
      "Nonremoveable", "Network drive", "CD-ROM", &
         "RAM disk" }

ls_driveletter = lb_drivelist.SelectedItem()
lui_drivetype = inv_filesrv.of_GetDriveType &
   (ls_driveletter)
MessageBox("File Service", "Drive type is: " &
   + ls_drivetype[lui_drivetype + 1] )
```

of_GetFileAttributes

Description

Retrieves all attributes for a specified file.

Access

Public

Syntax

instancename.**of_GetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *subdirectory*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose attributes are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
readonly	Boolean indicating whether *filename* is read-only (passed by reference)
hidden	Boolean indicating whether *filename* is hidden (passed by reference)
system	Boolean indicating whether *filename* is a system file (passed by reference)
subdirectory	Boolean indicating whether *filename* is a subdirectory (passed by reference)
archive	Boolean indicating whether *filename* has the archive bit set (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetFileAttributes function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_subdirectory.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
END IF
```

of_GetFileSize

Description

Retrieves the size of a specified file.

Access

Public

Syntax

instancename.**of_GetFileSize** (*filename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose size is to be returned. If you do not specify an absolute path, this function searches relative to the current working directory

Return value

Double. Returns the number of bytes in *filename* if the function succeeds and -1 if an error occurs.

Usage

This function calls the Findfirst and FindClose external functions.

Examples

This example calls the of_GetFileSize function (this example assumes an inv_filesrv instance variable):

```
Double   ld_return
String   ls_filename
```

```
ls_filename = sle_source.text
ld_return = &
   inv_filesrv.of_GetFileSize(ls_filename)
IF ld_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a size of " + String(ld_return) &
      + " bytes.")
END IF
```

of_GetLastAccessDate

Description

Retrieves the date a file was last accessed.

Access

Public

Syntax

instancename.**of_GetLastAccessDate** (*filename*, *date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last access date is to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the last access date (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetLastAccessDate function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date   ldt_date

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetLastAccessDate &
   (ls_filename, ldt_date)
IF li_return = -1 THEN
   MessageBox("File Service", &
```

```
        "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a last access date of " &
      + String(ldt_date) )
END IF
```

of_GetLastWriteDateTime

Description

Retrieves the date and time a file was last modified.

Access

Public

Syntax

instancename.**of_GetLastWriteDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last write date and time are to be returned. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date into which the function places the last write date (passed by reference)
time	Time into which the function places the last write time (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetLastWriteDateTime function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date   ldt_date
Time   lt_time

ls_filename = sle_source.text
li_return = inv_filesrv.of_GetLastWriteDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
```

```
    MessageBox("File Service", ls_filename &
       + " has a last write date of " &
       + String(ldt_date) &
       + " and a last write time of &
       " + String(lt_time) )
END IF
```

of_GetLongFilename

Description

Retrieves the corresponding long filename for a passed internal 8.3 (also called alternate) filename.

Access

Public

Syntax

instancename.**of_GetLongFilename** (*shortfilename*)

Argument	Description
instancename	Instance name of n_cst_filesrv
shortfilename	String specifying the short (alternate) filename

Return value

String. Returns the long filename for *shortfilename*.

Usage

This function calls the FindFirstFileA and FindClose external functions.

Examples

This example calls the of_GetLongFilename function (this example assumes an inv_filesrv instance variable):

```
String   ls_filename, ls_return

ls_filename = sle_source.text
ls_return = inv_filesrv.of_GetLongFilename &
   (ls_filename)
IF ls_return = "" THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
ELSE
   MessageBox("File Service", ls_return &
      + " is the long filename for " &
      + ls_filename)
END IF
```

of_RemoveDirectory

Description

Deletes the specified directory (the directory must be empty). To delete a directory and its files, call of_DelTree instead.

Access

Public

Syntax

instancename.**of_RemoveDirectory** (*directoryname*)

Argument	Description
instancename	Instance name of n_cst_filesrvwin
directoryname	String specifying the path to the directory to be deleted. If you do not specify an absolute path, this function deletes relative to the current working directory

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the RemoveDirectoryA external function.

Examples

This example checks if pbapp is the current directory and, if it is, deletes the pbapp\temp directory (this example assumes an inv_filesrv instance variable):

```
String   ls_return, ls_temp
Integer  li_return

ls_temp = "temp"
ls_return = &
   gnv_app.inv_platform.of_GetCurrentDirectory( )
IF Mid(ls_return, 4, 5) = "pbapp" THEN
   li_return = inv_filesrv.of_RemoveDirectory &
      (ls_temp)
   IF li_return = 1 THEN
      MessageBox("Delete", &
         "Temp dir does not exist")
   ELSEIF li_return <0 THEN
      MessageBox("Delete", &
         "Deletion error: Temp dir must be empty")
   END IF
...
```

of_SetCreationDateTime

Description

Sets a file's creation date and time.

Access

Public

Syntax

instancename.**of_SetCreationDateTime** (*filename*, *date*, *time*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose creation date and time are to be modified. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date variable containing the new creation date
time	Time variable containing the new creation time

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetCreationDateTime function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date   ldt_date
Time   lt_time

ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesrv.of_SetCreationDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
    "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a new creation date of " &
      + String(ldt_date) &
      + " and a new creation time of " &
      + String(lt_time) )
END IF
```

of_SetFileAttributes

Description

Sets attributes for the specified file.

Access

Public

Syntax

instancename.**of_SetFileAttributes** (*filename*, *readonly*, *hidden*, *system*, *archive*)

Argument	Description
instancename	Instance name of n_cst_filesrvwin32
filename	String specifying the file whose attributes are to be set. If you do not specify an absolute path, this function searches relative to the current working directory
readonly	Boolean indicating whether *filename* is read-only. If you specify NULL, the function doesn't modify this attribute
hidden	Boolean indicating whether *filename* is hidden. If you specify NULL, the function doesn't modify this attribute
system	Boolean indicating whether *filename* is a system file. If you specify NULL, the function doesn't modify this attribute
archive	Boolean indicating whether *filename* has the archive bit set. If you specify NULL, the function doesn't modify this attribute

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetFileAttributes function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename

ls_filename = sle_source.text
li_return = inv_filesrv.of_SetFileAttributes &
   (ls_filename, cbx_readonly.Checked, &
      cbx_hidden.Checked, &
      cbx_system.Checked, &
      cbx_archive.Checked)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error accessing " + ls_filename)
END IF
```

of_SetLastAccessDate

Description

Sets a file's last-accessed date and time.

Access

Public

Syntax

instancename.**of_SetLastAccessDate** (*filename, date*)

Argument	Description
instancename	Instance name of n_cst_filesrv
filename	String specifying the file whose last access date is to be modified. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date variable containing the new last-accessed date

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetLastAccessDate function (this example assumes an inv_filesrv instance variable):

```
Integer  li_return
String   ls_filename
Date   ldt_date

ls_filename = sle_source.text
ldt_date = Today()
li_return = inv_filesrv.of_SetLastAccessDate &
   (ls_filename, ldt_date)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a new last access date of " &
      + String(ldt_date))
END IF
```

of_SetLastWriteDateTime

Description

Sets a file's last-modified date and time.

Access

Public

Syntax

instancename.**of_SetLastWriteDateTime** (*filename, date, time*)

Argument	Description
instancename	Instance name of n_cst_filesrv

Argument	Description
filename	String specifying the file whose last write date and time are to be modified. If you do not specify an absolute path, this function searches relative to the current working directory
date	Date variable containing the new last-modified date
time	Time variable containing the new last-modified time

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetLastWriteDateTime function (this example assumes an inv_filesrv instance variable):

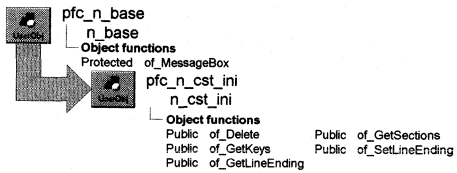
```
Integer  li_return
String   ls_filename
Date  ldt_date
Time  lt_time

ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesrv.of_SetLastWriteDateTime &
   (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
   MessageBox("File Service", &
      "Error modifying " + ls_filename)
ELSE
   MessageBox("File Service", ls_filename &
      + " has a new last write date of " &
      + String(ldt_date) &
      + " and a new last write time of " &
      + String(lt_time) )
END IF
```

n_cst_inifile

Description

INI file service. Provides functions to access and update INI files.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_filesrv
n_cst_string

Usage

To use this object:

1 Define an instance variable for n_cst_inifile. For MDI applications, this might be a frame-level instance variable (inv_ini_handler in this example), created in the frame window's Open event:

```
n_cst_inifile  inv_ini_handler
```

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements.

2 Use n_cst_inifile object functions as needed:

- Use of_GetKeys to retrieve all keys for a specific section
- Use of_GetSections to retrieve all section names in the INI file
- Use of_Delete to remove an entire section from the INI file

Using ProfileInt, ProfileString, and SetProfileString
Use the ProfileInt, ProfileString, and SetProfileString PowerScript functions to access INI file entries one at a time.

See also

n_cst_string

Events

N_cst_inifile contains a precoded event:

Constructor

Constructor

Description

Initializes the is_lineend instance variable with the platform-specific line end character or characters.

Usage

This event executes when the object is created.

Functions

N_cst_inifile contains object functions:

of_Delete	of_GetSections
of_GetKeys	of_SetLineEnding
of_GetLineEnding	

of_Delete

Description

Removes the specified entry or section from an INI file.

Access

Public

Syntax

instancename.**of_Delete** (*file*, *section* {, *key* })

Argument	Description
instancename	Instance name of n_cst_inifile
file	String naming the INI file containing the entry or section to be deleted
section	String naming the section to be deleted. If you also specify *key*, this value names the section containing the entry to be deleted
key (optional)	String naming the INI file entry to be deleted

Return value

Integer. Returns one of the following values:

1 Function succeeded; entry was deleted
0 Section does not exist within *file*
-1 File error
-2 *file* does not exist

Examples

This example removes the CustApp section:

```
Integer  li_return

li_return = inv_ini_handler.of_Delete &
   (gnv_app.of_GetAppINIFile(), "CustApp")
IF li_return <> 1 THEN
   MessageBox("Delete", "Entry not deleted")
END IF
```

of_GetKeys

Description

Retrieves all keys from a specified section.

Access

Public

Syntax

instancename.**of_GetKeys** (*file*, *sectionname*, *keynames*)

Argument	Description
instancename	Instance name of n_cst_inifile
file	String naming the INI file containing the section whose keys are retrieved
sectionname	String containing the name of the section whose keys are retrieved
keynames	String array to contain the section's key names (passed by reference)

Return value

Integer. Returns one of the following values:

- >0 The number of keys retrieved
- 0 Either there are no keys that exist for the section or the section does not exist
- -1 File error
- -2 The INI file has not been specified or does not exist

Usage

You can use the list of keys to retrieve all of the section's key values.

Examples

This example retrieves all the keys in the CustApp section and displays the list in the lb_keys ListBox:

```
String   ls_keys[ ]
Integer  li_count, li_size
```

```
li_size = inv_ini_handler.of_GetKeys &
   (gnv_app.of_GetAppINIFile(), "CustApp", ls_keys)
lb_keys.Reset()
FOR li_count = 1 to li_size
   lb_keys.AddItem(ls_keys[li_count])
NEXT
```

of_GetLineEnding

Description

Retrieves the platform-specific character or characters used to end lines in an INI file.

Access

Public

Syntax

instancename.**of_GetLineEnding ( )**

Argument	Description
instancename	Instance name of n_cst_inifile

Return value

String. Returns the line end character or characters.

Examples

This example calls the of_GetLineEnding function:

```
n_cst_inifile  lnv_ini
String  ls_lineend

ls_lineend = lnv_ini.of_GetLineEnding()
CHOOSE CASE ls_lineend
CASE "~n"
   MessageBox("Line Ending",  &
      "Newline - UNIX")
CASE "~r"
   MessageBox("Line Ending",  &
      "Carriage Return - Macintosh")
CASE "~r~n"
...
```

of_GetSections

Description

Retrieves all sections from an INI file.

Access

Public

Syntax

instancename.**of_GetSections** (*file*, *sectionnames*)

Argument	Description
instancename	Instance name of n_cst_inifile
file	String naming the INI file whose sections are retrieved
sectionnames	String array into which the function places the section names (passed by reference)

Return value

Integer. Returns one of the following values:

>0 The number of sections retrieved
-1 File error
-2 The INI file has not been specified or does not exist

Usage

You can then use the list of sections to retrieve a section's keys.

Examples

This example retrieves all the sections in the specified INI file and displays the list in the lb_sections ListBox:

```
String  ls_section[ ]
Integer li_count, li_size

li_size = inv_ini_handler.of_GetSections  &
   (gnv_app.of_GetAppINIFile(), ls_section)
lb_sections.Reset( )
FOR li_count = 1 to li_size
   lb_sections.AddItem(ls_section[li_count])
NEXT
```

of_SetLineEnding

Description

Specifies the platform-specific character or characters used to end lines in an INI file.

Access

Public

Syntax

instancename.**of_SetLineEnding** (*character*)

Argument	Description
instancename	Instance name of n_cst_inifile

Argument	Description
character	String containing the line end character or characters: ♦ **~n** Newline (UNIX) ♦ **~r** Carriage return (Macintosh) ♦ **~r~n** Carriage return - newline (Windows)

Return value

Integer. Returns 1 if the function succeeds -1 if an error occurs.

Examples

This example is from the n_cst_inifile Constructor event:

```
Environment  lenv_obj

GetEnvironment (lenv_obj)
if IsValid (lenv_obj) then
   choose case lenv_obj.ostype
      case aix!, osf1!, hpux!, aix!
         of_SetLineEnding("~n")
      case macintosh!
         of_SetLineEnding("~r")
      case else
         of_SetLineEnding("~r~n")
   end choose
else
   of_SetLineEnding ("~r~n")
end if
```

n_cst_luw

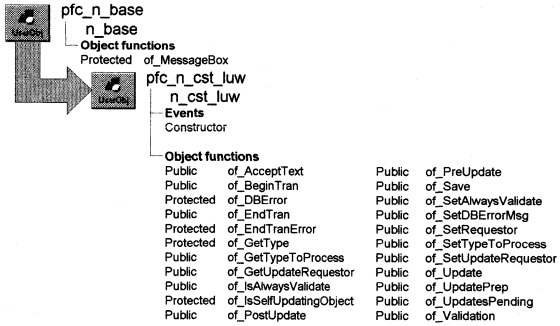
Description

Logical unit of work service. N_cst_luw provides support for self-updating objects. A self-updating object encapsulates all required update functionality by implementing a set of functions (self-updating object API) that n_cst_luw calls during the save process:

of_AcceptText
of_UpdatesPending
of_Validation
of_UpdatePrep
of_Update
of_PostUpdate

PFC includes several self-updating objects, including u_dw, n_ds, u_lvs, u_tab, u_base, w_master, and u_tvs. Refer to these objects to see implementations of the self-updating object API.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_dberrorattrib
n_cst_metaclass
n_tr

Usage

PFC uses n_cst_luw to coordinate and perform updates on a control or a group of controls. For example the w_master pfc_Save process calls n_cst_luw functions.

You can also use n_cst_luw as a standalone object to perform database updates via the of_Save function.

FOR INFO See the *PFC User's Guide*.

See also

n_ds

u_base
u_dw
u_lvs
u_tab
u_tvs
w_master

Instance variables

Instance variable	Description	Data type	Access	Usage
ALL_OBJECTS	Constant set to an empty string	String	Public	This is the default for ib_typetoprocess
ib_accepttext	Indicates whether the of_Save function accepts text by default	Boolean	Protected	The default is TRUE
ib_alwaysvalidate	Indicates whether n_cst_luw validates all controls, even those with no pending updates	Boolean	Protected	Set with of_SetAlwaysValidate (the default is FALSE)
ib_resetflag	Indicates whether the of_Save function resets update flags by default	Boolean	Protected	The default is FALSE
inv_dberrorattrib	Database error information	n_cst_dberrorattrib	Protected	Set with of_SetDBErrorMsg
inv_metaclass	Reference to metaclass service	n_cst_metaclass	Protected	Internal
ipo_pendingupdates[ ]	Controls with pending updates	PowerObject	Protected	Set by of_UpdatesPending
ipo_requestor	Object owning this instance of n_cst_luw	PowerObject	Protected	Set with of_Requestor
ipo_updaterequestor	Object requesting the update	PowerObject	Protected	Set with of_ SetUpdateRequestor
is_instancename	Unique name for this instance of n_cst_luw	String	Protected	Internal

Instance variable	Description	Data type	Access	Usage
is_typetoprocess	Customized list of object types to update	String	Protected	Set with of_SetTypeToProcess. Default is ALL_OBJECTS

Events

N_cst_luw includes a precoded event:

Constructor

Constructor

Description

Establishes a unique identifier for the current instance of n_cst_luw.

Usage

This event executes when the object is created.

Functions

N_cst_luw includes precoded functions:

of_AcceptText	of_PreUpdate
of_BeginTran	of_Save
of_DBError	of_SetAlwaysValidate
of_EndTran	of_SetDBErrorMsg
of_EndTranError	of_SetRequestor
of_GetType	of_SetTypeToProcess
of_GetTypeToProcess	of_SetUpdateRequestor
of_GetUpdateRequestor	of_Update
of_IsAlwaysValidate	of_UpdatePrep
of_IsSelfUpdatingObject	of_UpdatesPending
of_PostUpdate	of_Validation

of_AcceptText

Description

Calls the of_AcceptText function (if implemented) for all passed controls.

Access

Public

Syntax

instancename.**of_AcceptText** (*controlarray*, *focusonerror*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
controlarray	PowerObject array containing the control to check
focusonerror	Boolean specifying whether to set focus on a DataWindow column with errors

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

For non-PFC DataWindows and DataStores, this function calls the AcceptText function directly. For PFC DataWindows and other self updating objects, this function calls the object's of_AcceptText function, if implemented. To create additional accept text processing, extend the self-updating object's pfc_Accept event.

Examples

This example is from the w_master pfc_AcceptText event:

```
...
If IsValid(inv_luw) Then
   Return inv_luw.of_AcceptText(apo_control[ ],  &
      ab_focusonerror)
End If
...
```

of_BeginTran

Description

Begins transactions for the passed Transaction objects.

Access

Public

Syntax

instancename.**of_BeginTran** (*transactions*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
transactions	N_tr or n_tr array referencing Transaction objects for which start-of-transaction processing is initiated

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

To use this function:

- Optionally add code to the window's pfc_BeginTran event that calls this function
- Override the n_tr of_Begin function, adding code that performs begin transaction process as necessary

This function calls the n_tr of_Begin function for each object in the *transactions* array.

Examples

This example calls the of_BeginTran function:

```
n_tr  ltr_trans[ ]

ltr_trans[1] = SQLCA
IF IsValid(inv_luw) THEN
   IF inv_luw.of_BeginTran(ltr_trans) = -1 THEN
      Return -1
   END IF
END IF
Return 1
```

of_DBError

Description

Displays a message box using the passed database error message.

Access

Protected

Syntax

instancename.**of_DBError** (*errorinfo*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
errorinfo	N_cst_dberrorattrib containing database error information

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Save function:

```
...
If li_save_rc<=0 Then
  This.of_dberror(inv_dberrorattrib)
...
```

of_EndTran

Description

Commits or rolls back all specified transactions as appropriate.

Access

Public

Syntax

instancename.**of_EndTran** (*transactions*, *savestate*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
transactions	N_tr array containing transactions
savestate	Integer containing the current status of the save process

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Save function:

```
...
li_endtran_rc =  &
   This.of_endtran(atr_control, li_save_rc)
...
```

of_EndTranError

Description

Displays a message box listing end of transaction errors. This can include information on transactions that were committed and those that were rolled back.

Access

Protected

Syntax

instancename.**of_EndTranError** (*trcommit*, *trrollback*, *returncodes*, *savestate*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
trcommit	N_tr array referencing transactions that were committed
trrollback	N_tr array referencing transactions that were rolled back
returncodes	Integer array containing ROLLBACK return codes
savestate	Integer containing the current status of the save process

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_EndTran function:

```
...
li_rc = of_EndTranError(ltr_performed,  &
   ltr_secondary, li_secondaryrc, ai_saverc)
...
```

of_GetType

Description

Retrieves the type of the specified control.

Access

Protected

Syntax

instancename.**of_GetType** (*control*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
control	PowerObject containing the control

Return value

String. Returns the control name if the function succeeds and ! if an error occurs.

Usage

Internal.

Examples

This example is from the of_AcceptText function:

```
...
If Pos(is_typetoprocess,  &
   of_GetType(lpo_tocheck)) = 0 Then
      Continue
End If
...
```

of_GetTypeToProcess

Description

Retrieves the types of objects accessed by the save process.

Access

Public

Syntax

instancename.**of_GetTypeToProcess** ( )

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)

Return value

String. Returns the types of objects accessed by the save process (an empty string means all objects are supported).

Examples

This example calls the of_GetTypeToProcess function:

```
String  ls_types

ls_types = parent.inv_luw.of_GetTypeToProcess()
MessageBox("LUW", "Save types are " + ls_types)
```

of_GetUpdateRequestor

Description

Retrieves a reference to the object requesting the update.

Access

Public

Syntax

instancename.**of_GetUpdateRequestor** (*requestor*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
requestor	PowerObject into which the function places a reference to the object requesting the update (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, 0 if there is no update requestor, and -1 if an error occurs.

Usage

An update requestor, which establishes an owner for self-updating objects, is optional.

Examples

This example calls the of_GetUpdateRequestor function:

```
PowerObject  lpo_req

IF inv_luw.of_GetUpdateRequestor(lpo_req) > 1 THEN
   MessageBox("LUW", "Update requestor is: "  &
      + lpo_req.ClassName())
END IF
```

of_IsAlwaysValidate

Description

Reports whether n_cst_luw always performs validation.

Access

Public

Syntax

instancename.**of_IsAlwaysValidate** ( )

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)

Return value

Boolean. Returns TRUE if n_cst_luw performs validation on all objects (event those without pending updates) and FALSE if n_cst_luw performs validation on objects with pending updates only.

Examples

This example calls the of_IsAlwaysValidate function:

```
IF parent.inv_luw.of_IsAlwaysValidate() = TRUE THEN
   MessageBox("LUW", "Always validate")
ELSE
   MessageBox("LUW", "Sometimes validate")
END IF
```

of_IsSelfUpdatingObject

Description

Reports whether the passed control is a self-updating object.

Access

Protected

Syntax

instancename.**of_IsSelfUpdatingObject** (*control*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
control	PowerObject containing the control

Return value

Boolean. Returns TRUE if *control* is a self updating object.

Usage

Override this function redefine the functions that make up a self-updating object.

Examples

This example is from the of_AcceptText function:

```
...
Case Window!
   If Not of_IsSelfUpdatingObject(lpo_tocheck) Then
```

```
            lw_control = lpo_tocheck
            li_rc = This.of_AcceptText  &
               (lw_control.control, ab_FocusOnError )
...
```

of_PostUpdate

Description

Calls the of_PostUpdate function (if implemented) for all passed controls.

Access

Public

Syntax

instancename.**of_PostUpdate** (*controls*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
controls	PowerObject array containing the controls for which DataWindow update flags are reset

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

For non-PFC DataWindows, this function calls the ResetUpdate function directly. For PFC DataWindows and other self-updating objects, this function calls the object's of_PostUpdate function, which resets updates flags as appropriate. To create additional post-update processing, extend the self-updating object's pfc_PostUpdate event.

Examples

This example calls the of_PostUpdate function:

```
...
If IsValid(inv_luw) Then
   Return inv_luw.of_PostUpdate(apo_control)
End If
...
```

of_PreUpdate

Description

Empty function that you override in n_cst_luw to perform application-specific pre-update processing.

Access

Public

Syntax

instancename.**of_PreUpdate** ( )

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the w_master pfc_PreUpdate event:

```
...
If IsValid(inv_luw) Then
   Return inv_luw.of_PreUpdate()
End If
...
```

of_Save

Description

Performs a complete save operation for the specified object or objects.

Access

Public

Syntax

instancename.**of_Save** (*control, transaction*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
control	PowerObject or PowerObject array containing the control or controls to be updated
transaction	N_tr or n_tr array containing the Transaction objects required by the update

Return value

Integer. Returns values as follows:

- **1** Success
- **0** No pending updates
- **-1** Accept text error
- **-2** Updates pending error
- **-3** Validation error
- **-4** Pre-update error
- **-5** Begin transaction error
- **-6** Update error

- **-7** End transaction error
- **-8** Post save error
- **-9** Update prep error

Usage

Call this function to perform the complete save process using n_cst_luw only.

The *transaction* argument is used by begin transaction and end transaction processing.

This function calls other n_cst_luw functions:

of_AcceptText
of_UpdatesPending
of_Validation
of_UpdatePrep
of_PreUpdate
of_BeginTran
of_Update
of_DBError (as appropriate)
of_PostUpdate

Examples

This example calls the of_Save function:

```
n_cst_luw  lnv_luw
Integer  li_return

lnv_luw = CREATE n_cst_luw
li_return = lnv_luw.of_Save(tv_1, SQLCA)
IF li_return < 0 THEN
   MessageBox("Save", "Save Error")
END IF
DESTROY lnv_luw
```

of_SetAlwaysValidate

Description

Specifies whether n_cst_luw performs validation even for those objects that do not have pending updates.

Access

Public

Syntax

instancename.**of_SetAlwaysValidate** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)

Argument	Description
boolean	Boolean specifying whether n_cst_luw always perform validation (TRUE) or only performs validation if a control has pending updates

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetAlwaysValidate function:

```
inv_luw.of_SetAlwaysValidate(TRUE)
```

of_SetDBErrorMsg

Description

Specifies the message that appears for a database error.

Access

Public

Syntax

instancename.**of_SetDBErrorMsg** (*errorinfo*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
errorinfo	N_cst_dberrorattrib containing database error information

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The message specified here is displayed by the of_DBError function.

Examples

This example calls the of_SetDBErrorMsg function:

```
n_cst_dberrorattrib lnv_dberrorattrib

inv_dberrorattrib.is_errormsg = "Update Error"
inv_luw.of_SetDBErrorMsg(lnv_dberrorattrib)
```

of_SetRequestor

Description

Associates the requesting object with this instance of n_cst_luw.

Access

Public

Syntax

instancename.**of_SetRequestor** (*requestor*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
requestor	PowerObject containing the requesting object (typically a w_master, u_tab, or u_base descendant)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This reference is not required.

Examples

This example is from the w_master of_SetLogicalUnitOfWork function:

```
...
if IsNull(inv_luw) Or not IsValid(inv_luw) then
      inv_luw = create n_cst_luw
      inv_luw.of_SetRequestor(this)
      inv_luw.of_SetUpdateRequestor(this)
      return 1
   end if
...
```

of_SetTypeToProcess

Description

Specifies the types of controls accessed by a default save process.

Access

Public

Syntax

instancename.**of_SetTypeToProcess** (*controltypes*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
controltypes	String containing a blank separated list of the controls processed by a default save process

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The default is an empty string (or the ALL_OBJECTS constant), which results in all controls being processed.

Examples

This example calls the of_SetTypeToProcess function:

```
inv_luw.of_SetTypeToProcess  &
   ("datawindow listview treeview")
```

of_SetUpdateRequestor

Description

Associates n_cst_luw with the object requesting the update process.

Access

Public

Syntax

instancename.**of_SetUpdateRequestor** (*requestor*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
requestor	PowerObject containing the updating object or control

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

When appropriate, n_cst_luw uses this reference when referencing self-updating objects. If you don't set this reference, n_cst_luw uses itself as the reference.

Examples

This example calls the of_SetUpdateRequestor function:

```
...
if IsNull(inv_luw) Or not IsValid(inv_luw) then
      inv_luw = create n_cst_luw
      inv_luw.of_SetRequestor(this)
      inv_luw.of_SetUpdateRequestor(this)
      return 1
   end if
...
```

of_Update

Description

Calls the of_Update function (if implemented) for all passed controls.

Access

Public

Syntax

instancename.**of_Update** (*controlarray*, *accepttext*, *resetflags*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
controlarray	PowerObject array containing the controls to be updated
accepttext	Boolean indicating whether to perform an accept text before performing the update
resetflags	Boolean indicating whether to automatically reset update flags

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

For non-PFC DataWindows and DataStores, this function calls the Update function directly. For PFC DataWindows and other self-updating objects, this function calls the object's of_Update function, if implemented. To create additional update processing, extend the self-updating object's pfc_Update event.

Examples

This example is from the w_master pfc_UpdateObjects event:

```
...
If IsValid(inv_luw) Then
   Return inv_luw.of_Update(apo_control, &
      ab_accepttext, ab_resetflag)
End If
...
```

of_UpdatePrep

Description

Calls the of_UpdatePrep function (if implemented) for all passed controls.

Access

Public

Syntax

instancename.**of_UpdatePrep** (*controlarray*)

Argument	Description
instancename	Instance name of n_cst_luw (the PFC default is inv_luw)
controlarray	PowerObject array containing the controls to be updated

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

For non-PFC DataWindows and DataStores, this function calls the ue_UpdatePrep event, which you extend in the self-updating object to perform customized update preparation. For PFC DataWindows and other self-updating objects, this function calls the object's of_UpdatePrep function, if implemented. To create additional update processing, extend the self-updating object's pfc_UpdatePrep event

Examples

This example is from the w_master pfc_UpdatePrep event:

```
...
If IsValid(inv_luw) Then
   Return inv_luw.of_UpdatePrep(apo_control)
End If
```

```
...
```

of_UpdatesPending

Description

Calls the of_UpdatePending function (if implemented) for all passed controls.

Access

Public

Syntax

instancename.**of_UpdatesPending** (*controlarray*, *pendingarray*)

Argument	Description
instancename	Instance name of n_cst_luw (the w_master default is inv_luw)
controlarray	PowerObject array containing the controls to be updated
pendingarray	PowerObject array into which the function places a list of controls that have pending updates (passed by reference)

Return value

Integer. Returns 1 if updates are pending, 0 if no updates are pending, and -1 if an error occurs.

Usage

For non-PFC DataWindows and DataStores, this function accesses the objects directly to determine whether there are pending updates. For PFC DataWindows and other self-updating objects, this function calls the object's of_UpdatesPending function, if implemented. To create additional update processing, extend the self-updating object's pfc_UpdatesPending event.

There is also a Protected version of this function, which is for internal use.

Examples

This example is from the w_master pfc_UpdatesPendingRef event:

```
...
If IsValid(inv_luw) Then
   li_rc = inv_luw.of_UpdatesPending(apo_control,  &
     lpo_pending)
   apo_pending = lpo_pending
   Return li_rc
End If
...
```

of_Validation

Description

Calls the of_Validation function (if implemented) for all passed controls.

Access

Public

Syntax

instancename.**of_Validation** (*controlarray*)

Argument	Description
instancename	Instance name of n_cst_luw (the w_master default is inv_luw)
controlarray	PowerObject array containing controls to be validated

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

For non-PFC DataWindows and DataStores, this function calls the ue_validation function, which you can optionally implement in these controls. For PFC DataWindows and other self-updating objects, this function calls the object's of_Validation function, if implemented. To create additional update processing, extend the self-updating object's pfc_Validation event

Examples

This example is from the w_master pfc_Validation event:

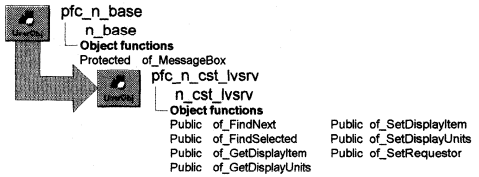
```
...
If IsValid(inv_luw) Then
   Return inv_luw.of_Validation(apo_control)
End If
...
```

n_cst_lvsrv

Description

ListView base service.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_ds
u_lvs

Usage

To use this service:

1 Enable the service by calling the u_lvs of_SetBase function:

```
lv_1.of_SetBase(TRUE)
```

2 Populate the ListView.

3 Call n_cst_lvsrv functions as needed.

Descendants

n_cst_lvsrv_datasource
n_cst_lvsrv_sort

See also

u_lvs
u_tvs

Instance variables

Instance variable	Description	Data type	Access	Usage
is_displayitem	Controls message text displayed for a single row	String	Protected	Call of_SetDisplayItem and of_GetDisplayItem to access
is_displayunits	Controls message text displayed for multiple rows	String	Protected	Call of_SetDisplayUnits and of_GetDisplayUnits to access

Instance variable	Description	Data type	Access	Usage
ilv_requestor	ListView control that owns the n_cst_lvsrv instance	u_lvs	Protected	Used to access and control the associated ListView control

Functions

N_cst_lvsrv includes precoded functions:

of_FindNext	of_SetDisplayItem
of_FindSelected	of_SetDisplayUnits
of_GetDisplayItem	of_SetRequestor
of_GetDisplayUnits	

of_FindNext

Description

Searches for the next item relative to a specific location in the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_FindNext** (*startindex, direction*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
startindex	Integer specifying the index number from which you want your search to begin
direction	Direction enumerated data type specifying the search direction: DirectionAll! DirectionUp! DirectionDown! DirectionLeft! DirectionRight!

Return value

Integer. Returns the index of the item found if the function succeeds and -1 if an error occurs.

Usage

If you want to search from the beginning, specify 0 for *startindex*.

This function does not select the item it finds. Use the item's Selected property in conjunction with this function to select the resulting match.

Examples

This example calls the of_FindNext function:

```
listviewitem   l_lvi
integer        li_index li_startindex

li_startindex = lv_list.SelectedIndex( )
lv_list.inv_base.of_FindNext(li_startindex,  &
   DirectionDown!)
IF li_index = -1 THEN
   MessageBox("Error", "Item not found.")
ELSE
   lv_list.GetItem (li_index, l_lvi )
   l_lvi.HasFocus = TRUE
   l_lvi.Selected = TRUE
   lv_list.SetItem(li_index,l_lvi)
END IF
```

of_FindSelected

Description

Searches for the next ListView item that is selected.

Access

Public

Syntax

lvcontrol.instancename.**of_FindSelected** (*startindex*, *direction*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
startindex	Integer specifying the index number from which you want your search to begin
direction	Direction enumerated data type specifying the search direction: DirectionAll! DirectionUp! DirectionDown! DirectionLeft! DirectionRight!

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If you want to search from the beginning, specify 0 for *startindex*.

Examples

This example calls the of_FindSelected function:

```
...
lv_list.inv_base.of_FindSelected(0,  &
   DirectionDown!)
```

of_GetDisplayItem

Description

Retrieves the display name for single items in the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_GetDisplayItem ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)

Return value

String. Returns the display name for single items in the ListView.

Usage

PFC uses this name in message text when confirming single-item deletion. For example, if this name is *this row*, PFC displays Are you sure you want to delete this row?

Examples

This example calls the of_GetDisplayItem function:

```
String    ls_item

ls_item = lv_1.inv_base.of_GetDisplayItem( )
MessageBox("ListView", "Item label is: " &
   + ls_item)
```

of_GetDisplayUnits

Description

Retrieves the display name for multiple rows in the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_GetDisplayUnits ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)

Return value

String. Returns the display name for multiple items in the ListView.

Usage

PFC uses this name in message text when confirming multiple item deletion. For example, if this name is *rows*, PFC displays Are you sure you want to delete these rows?

Examples

This example calls the of_GetDisplayUnits function:

```
String    ls_units

ls_units = lv_1.inv_base.of_GetDisplayUnits( )
MessageBox("ListView", "Unit label is: " &
   + ls_units)
```

of_SetDisplayItem

Description

Specifies the display name for single items in the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_SetDisplayItem** (*itemname*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
itemname	String specifying the display name for single items in the ListView

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

PFC uses this name in message text when confirming single-row deletion. For example, if you specify *this row*, PFC displays Are you sure you want to delete this row?

Examples

This example calls the of_SetDisplayItem function:

```
...
```

```
lv_1.inv_base.of_SetDisplayItem("the selected row")
...
```

of_SetDisplayUnits

Description

Specifies the display name for multiple items in the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_SetDisplayUnits** (*units*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
units	String specifying the display name for multiple rows of the ListView

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

PFC uses this name in message text when confirming multiple item deletion. For example, if this name is *rows*, PFC displays Are you sure you want to delete these rows?

Examples

This example calls the of_SetDisplayUnits function:

```
...
lv_1.inv_base.of_SetDisplayUnits("selected rows")
...
```

of_SetRequestor

Description

Associates the requesting object with this instance of n_cst_lvsrv.

Access

Public

Syntax

lvcontrol.instancename.**of_SetRequestor** (*requestor*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)

Argument	Description
requestor	U_lvs-based ListView control to be associated with the service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_lvs_of_SetBase function:

```
...
if ab_Switch then
   if IsNull(inv_base) Or not IsValid (inv_base) then
      inv_base = create n_cst_lvsrv
      inv_base.of_SetRequestor(this)
      li_rc = 1
   end if
...
```

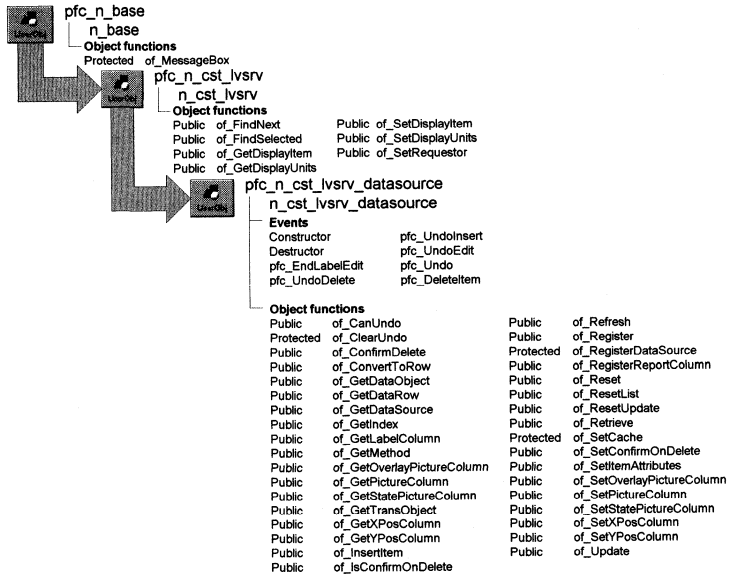
n_cst_lvsrv_datasource

Description

ListView data access object. Use this object to display data in a u_lvs control and perform actions on that data. This data can come from a:

SQL statement
DataWindow object
DataStore
DataWindow control
PowerObject

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_dwcache
n_ds
n_tr

Usage

To use this object:

1 Enable the service using the u_lvs of_SetDataSource function:

```
this.of_SetDataSource(TRUE)
```

2 Define a data source by calling the of_Register function:

```
this.inv_datasource.of_Register("emp_lname",  &
```

```
"d_emplist", SQLCA)
```

3 Call additional functions to control ListView behavior:

```
this.inv_datasource.of_RegisterReportColumn()
this.inv_datasource.of_SetPictureColumn("1")
```

4 Populate the ListView by calling the u_lvs pfc_Populate event:

```
this.Event pfc_Populate()
```

See also

n_cst_lvsrv_sort
u_lvs

Instance variables

Instance variable	Description	Data type	Access	Usage
CACHE_ID	Constant set to "pfc listview"	String	Public	Internal
ib_confirmondelete	Indicates whether confirm on delete is enabled	Boolean	Protected	Set with of_SetConfirmOnDelete
ib_isrefreshing	Indicates whether a refresh is occurring	Boolean	Protected	Internal
ib_isresetting	Indicates whether a reset is occurring	Boolean	Protected	Internal
ib_isundoing	Indicates whether an undo is occurring	Boolean	Protected	Internal
ids_source	Data source for the ListView	n_ds	Protected	Set with of_Register
il_undodeletehandle[ ]	Deleted items	Long	Protected	Internal
il_undoedithandle	Modified items	Long	Protected	Internal
il_undoinserthandle	Last inserted item	Long	Protected	Internal
ilvi_undodeleteitem[ ]	Deleted items	ListViewItem	Protected	Internal
inv_cache	Cached data	n_cst_dwcache	Protected	Internal
is_dataobject	DataWindow object used to populate the ListView	String	Protected	Set with of_Register

Instance variable	Description	Data type	Access	Usage
is_keycolumns[ ]	Key columns	String	Protected	Set with of_Register
is_labelcolumn	Label column	String	Protected	Set with of_Register
is_undoeditlabeltext	Modified label text	String	Protected	Internal
is_undotype	Undo type	String	Protected	Internal
is_xposcolumn	Column specifying x position	String	Protected	Set with of_SetXPosColumn
is_yposcolumn	Column specifying x position	String	Protected	Set with of_SetYPosColumn
itr_obj	Transaction object	n_tr	Protected	Set with of_Register
NO_ACTION	Constant set to 0	Integer	Public	Internal
PREVENT_ACTION	Constant set to 0	Integer	Public	Internal
UNDO_DELETE	Constant set to "Delete"	String	Public	Internal
UNDO_EDIT	Constant set to "Edit"	String	Public	Internal
UNDO_INSERT	Constant set to "Insert"	String	Public	Internal

Events

N_cst_lvsrv_datasource includes precoded events:

Constructor	pfc_Undo
Destructor	pfc_UndoDelete
pfc_DeleteItem	pfc_UndoEdit
pfc_EndLabelEdit	pfc_UndoInsert

Constructor

Description — Enables the caching service.

Usage — This event executes when the object is created.

Destructor

Description — Destroys the caching service.

Usage

This event executes when the object is created.

pfc_DeleteItem

Description

Deletes a row from the DataStore.

Syntax

lvcontrol.instancename.EVENT **pfc_DeleteItem** (*index*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)
index	Integer specifying the position of the ListView item to delete

Return value

Integer. Returns 1 if the event succeeds and 0 if no action is taken.

Usage

This event does not perform processing if refreshing, resetting, or undoing an insert.

Examples

This example is from the u_lvs DeleteItem event:

```
...
If isvalid(inv_datasource) then
   inv_datasource.event pfc_deleteitem(index)
...
```

pfc_EndLabelEdit

Description

Change the label column in the associated data source.

Syntax

lvcontrol.instancename.EVENT **pfc_EndLabelEdit** (*index, newlabel*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)
index	Integer specifying the index of the modified ListView item
newlabel	String specifying the updated label

Return value

Integer. Returns 0 if the label update is allowed and 1 if PowerBuilder should prevent the label update.

Usage

PFC allows label update only when the associated label column is updatable. If the label column is a computed column, you must override this event to update the computed column and return a 0.

This event does not update the database.

Examples

This example is from the u_lvs EndLabelEdit event:

```
...
If IsValid(inv_datasource) Then
   li_rc = inv_datasource.event pfc_EndLabelEdit  &
      (index, newlabel)
   If li_rc < 1 then Return 1
End If
...
```

pfc_Undo

Description

Restores the last ListView change.

Syntax

lvcontrol.instancename.EVENT **pfc_Undo ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)

Return value

Integer. Returns 1 if the event succeeds, 0 if there is nothing to undo, and -1 if an error occurs.

Examples

This example is from the u_lvs pfc_Undo event:

```
...
if isvalid(inv_datasource) then
   li_rc = inv_datasource.event pfc_undo()
end if
...
```

pfc_UndoDelete

Description

Restores the last deleted item (or items if multiple items were deleted at once).

Syntax

lvcontrol.instancename.EVENT **pfc_UndoDelete ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)

Return value

Integer. Returns 1 if the event succeeds, 0 if there is nothing to undo, and -1 if an error occurs.

Examples

This example is from the pfc_Undo event:

```
...
Choose Case ls_UndoType
   Case UNDO_DELETE
      return this.event pfc_undodelete()
   Case UNDO_INSERT
      return this.event pfc_undoinsert()
   Case UNDO_EDIT
      return this.event pfc_undoedit()
   End Choose
...
```

pfc_UndoEdit

Description

Restores a label's previous value.

Syntax

lvcontrol.instancename.EVENT **pfc_UndoEdit ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)

Return value

Integer. Returns 1 if the event succeeds, 0 if there is nothing to undo, and -1 if an error occurs.

Examples

This example is from the pfc_Undo event:

```
...
Choose Case ls_UndoType
   Case UNDO_DELETE
      return this.event pfc_undodelete()
   Case UNDO_INSERT
```

```
        return this.event pfc_undoinsert()
    Case UNDO_EDIT
        return this.event pfc_undoedit()
    End Choose
...
```

pfc_UndoInsert

Description

Removes the last inserted item.

Syntax

lvcontrol.instancename.EVENT **pfc_UndoInsert** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Examples

This example is from the pfc_Undo event:

```
...
Choose Case ls_UndoType
    Case UNDO_DELETE
        return this.event pfc_undodelete()
    Case UNDO_INSERT
        return this.event pfc_undoinsert()
    Case UNDO_EDIT
        return this.event pfc_undoedit()
    End Choose
...
```

Functions

N_cst_lvsrv_datasource includes precoded functions:

of_CanUndo	of_Refresh
of_ClearUndo	of_Register
of_ConfirmDelete	of_RegisterDataSource
of_ConvertToRow	of_RegisterReportColumn
of_GetDataObject	of_Reset

of_GetDataRow	of_ResetList
of_GetDataSource	of_ResetUpdate
of_GetIndex	of_Retrieve
of_GetLabelColumn	of_SetCache
of_GetMethod	of_SetConfirmOnDelete
of_GetOverlayPictureColumn	of_SetItemAttributes
of_GetPictureColumn	of_SetOverlayPictureColumn
of_GetStatePictureColumn	of_SetPictureColumn
of_GetTransObject	of_SetStatePictureColumn
of_GetXPosColumn	of_SetXPosColumn
of_GetYPosColumn	of_SetYPosColumn
of_InsertItem	of_Update
of_IsConfirmOnDelete	

of_CanUndo

Description

Reports whether the ListView can undo the last edit, insertion, or deletion, optionally returning the operation type.

Access

Public

Syntax

lvcontrol.instancename.**of_CanUndo** ({ *undotype* })

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
undotype (optional)	String into which the function places the type of the last operation (passed by reference): ♦ **UNDO_EDIT or "Edit"** Edit ♦ **UNDO_INSERT or "Insert"** Insertion ♦ **UNDO_DELETE or "Delete"** Deletion

Return value

Boolean. Returns TRUE if the last operation can be undone and FALSE if it cannot.

Examples

This example calls the of_CanUndo function:

```
IF lv_1.inv_datasource.of_CanUndo() THEN
   MessageBox("ListView", "Can undo")
ELSE
   MessageBox("ListView", "Cannot undo")
END IF
```

of_ClearUndo

Description

Clears the ListView's undo properties.

Access

Protected

Syntax

lvcontrol.instancename.**of_ClearUndo** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Refresh function:

```
...
this.of_clearundo()
...
```

of_ConfirmDelete

Description

Displays a message box allowing the user to confirm a deletion.

Access

Public

Syntax

lvcontrol.instancename.**of_ConfirmDelete** (*amount*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
amount	Long specifying the number of rows to be deleted

Return value

Integer. Returns 1 if deletion should proceed, 2 if deletions should not proceed, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the n_cst_lvsrv_datasource of_DeleteItem function:

```
...
li_confirm = inv_datasource.of_ConfirmDelete(1)
...
```

of_ConvertToRow

Description

Converts an array of data into a row in the DataStore.

Access

Public

Syntax

lvcontrol.instancename.**of_ConvertToRow** (*colvalues*, *datastore*, *row*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
colvalues	Any array containing the data to be used for the new row
datastore	N_ds instance into which the function places the new row (passed by reference)
row	Long into which the function places the new row number (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_lvs of_InsertItem function:

```
...
If inv_datasource.of_ConvertToRow  &
  (aa_columnvalues, lds_datastore, ll_row) <> 1 Then
...
```

of_FormatData

Description

Converts DataWindow data from its native format to the string data type.

Access

Public

Syntax

lvcontrol.instancename.**of_FormatData** (*columnname*, *format*, *type*, *datastore*, *row*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control

Argument	Description
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
columnname	String specifying the column
format	String specifying the column's format
type	String specifying the column's data type
datastore	N_ds specifying the DataStore containing the data
row	Long specifying the row for which the function returns column data

Return value

String. Returns the specified row/column data as a formatted string.

Examples

This example calls the of_FormatData function:

```
Integer  li_cols, li_cnt
String  ls_column, ls_label
...
li_cols = UpperBound(is_Columns)
FOR li_cnt = 1 TO li_cols
   ls_column = of_FormatData(is_columns[li_cnt], &
      is_colformat[li_cnt], is_coltype[li_cnt], &
         ids_data, al_row)
...
```

of_GetColumnInfo

Description

Retrieves column information from the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_GetColumnInfo** (*column*, *colattrib*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
column	Integer specifying the column from the ListView
colattrib	N_cst_columnattrib into which the function places column information (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Column information is used in detail view only.

Examples

This example is from the n_cst_lvsrv_sort pfc_ColumnClick event:

```
...
of_GetColumnInfo(ai_column, lnv_colinfo)
...
```

of_GetDataObject

Description

Retrieves the DataWindow object used as the ListView's data source.

Access

Public

Syntax

lvcontrol.instancename.**of_GetDataObject** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the name of the DataWindow object used to store data for display in the ListView.

Examples

This example calls the of_GetDataObject function:

```
String   ls_dataobject

ls_dataobject = lv_1.inv_datasource.of_GetDataObject()
MessageBox("ListView", "DataWindow object is: "  &
   + ls_dataobject)
```

of_GetDataRow

Description

Retrieves from the DataStore the row that corresponds to the specified ListView item.

Access

Public

Syntax

lvcontrol.instancename.**of_GetDataRow** (*item*, *datasource*, *row*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
item	Integer specifying the index to the ListView item
datasource	N_ds specifying the DataStore that contains the row (passed by reference)
row	Long specifying the row in *datasource* that corresponds to *item* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Use this function to update rows in the data source.

Examples

This example is from the pfc_UndoEdit event:

```
...
If of_GetDataRow(ll_undohandle, lds_DataStore,  &
   ll_Row) = -1 Then
      Return -1
End If
...
```

of_GetDataSource

Description

Retrieves the DataStore used as the source for the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_GetDataSource** (*source*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
source	N_ds instance into which the function places a reference to the DataStore used as the source for the ListView (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Use *source* to access the ListView's internal data source.

Examples

This example is from the u_lvs pfc_Refresh event:

```
...
inv_datasource.of_getdatasource(lds_source)
return this.event pfc_AddAll(lds_source)
...
```

of_GetIndex

Description

Retrieves the ListView item index for the specified row.

Access

Public

Syntax

lvcontrol.instancename.**of_GetIndex** (*row*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
row	Long specifying the DataStore row for which the function returns the ListView item index

Return value

Integer. Returns the index of the associated ListView item if the function succeeds, 0 if the item was not found, and -1 if an error occurs.

Examples

This example calls the of_GetIndex function:

```
Integer  li_item

li_item = lv_1.inv_datasource.of_GetIndex(10)
MessageBox("ListView", "Row 10 is item: "  &
   + String(li_item)
```

of_GetLabelColumn

Description

Retrieves the name of the column used as the ListView item label.

Access

Public

Syntax

lvcontrol.instancename.**of_GetLabelColumn** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the data source column name.

Examples

This example calls the of_GetLabelColumn function:

```
String  ls_col

ls_col = lv_1.inv_datasource.of_GetLabelColumn()
MessageBox("ListView", "Label column is: "  &
   + ls_col)
```

of_GetMethod

Description

Retrieves the method used to populate the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_GetMethod ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the method used to populate the Listview:

DataObjectData
DataStoreControl
DataWindowControl
ImportFile
PowerObject
Retrieve
SQL

Usage

The of_Register function controls the method used to populate the ListView.

Examples

This example calls the of_GetMethod function:

```
String  ls_method
```

```
ls_method = lv_1.inv_datasource.of_GetMethod()
MessageBox("ListView", "Method is: " + ls_method)
```

of_GetOverlayPictureColumn

Description

Retrieves the name of the column that specifies the overlay picture.

Access

Public

Syntax

lvcontrol.instancename.**of_GetOverlayPictureColumn ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the name of the column that specifies the overlay picture.

Examples

This example calls the of_GetOverlayPictureColumn function:

```
String  ls_column

ls_column =  &
   lv_1.inv_datasource.of_GetOverlayPictureColumn()
MessageBox("ListView", "Overlay is: " + ls_column)
```

of_GetPictureColumn

Description

Retrieves the name of the column that specifies the picture used in icon view.

Access

Public

Syntax

lvcontrol.instancename.**of_GetPictureColumn ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the name of the column that specifies the picture used in icon view.

Examples

This example calls the of_GetPictureColumn function:

```
String  ls_column

ls_column = lv_1.inv_datasource.of_GetPictureColumn()
MessageBox("ListView", "Picture is: " + ls_column)
```

of_GetStatePictureColumn

Description

Retrieves the name of the column that specifies the state picture.

Access

Public

Syntax

lvcontrol.instancename.**of_GetStatePictureColumn** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the name of the column that specifies the state picture.

Examples

This example calls the of_GetStatePictureColumn function:

```
String  ls_column

ls_column =  &
   lv_1.inv_datasource.of_GetStatePictureColumn()
MessageBox("ListView", "State picture is: " +  &
   ls_column)
```

of_GetTransObject

Description

Retrieves the Transaction object used to display data in the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_GetTransObject** (*transaction*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Argument	Description
transaction	N_tr into which the function places a reference to the Transaction object used to display ListView data (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The of_Register function specifies the Transaction object. Not all data sources require a Transaction object.

Examples

This example calls the of_GetTransObject function:

```
n_tr  ltr_trans

lv_1.inv_datasource.of_GetTransObject(ltr_trans)
IF IsValid(ltr_trans) THEN
   ltr_trans.of_Commit()
END IF
```

of_GetXPosColumn

Description

Retrieves the name of the column that specifies the item's x coordinate.

Access

Public

Syntax

lvcontrol.instancename.**of_GetXPosColumn ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the name of the column that specifies the item's x coordinate.

Examples

This example calls the of_GetXPosColumn function:

```
String  ls_column

ls_column = lv_1.inv_datasource.of_GetXPosColumn()
MessageBox("ListView", "XPos column is: "  &
   + ls_column)
```

of_GetYPosColumn

Description

Retrieves the name of the column that specifies the item's y coordinate.

Access

Public

Syntax

lvcontrol.instancename.**of_GetYPosColumn** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

String. Returns the name of the column that specifies the item's y coordinate.

Examples

This example calls the of_GetYPosColumn function:

```
String  ls_column

ls_column = lv_1.inv_datasource.of_GetYPosColumn()
MessageBox("ListView", "YPos column is: " &
   + ls_column)
```

of_InsertItem

Description

Adds a new item to the ListView using a row from the data source.

Access

Public

Syntax

lvcontrol.instancename.**of_InsertItem** (*datastore*, *row*, *lvitem*, *position*, *item*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
datastore	N_ds containing the data for the new item. This can be either the DataStore maintained by n_cst_lvsrv_datasource or another DataStore. If this is another DataStore, the associated DataWindow objects must be the same and the function also adds the row to the DataStore maintained by n_cst_lvsrv_datasource
row	Long specifying the row containing the new item
lvitem	ListViewItem containing the item to be added to the ListView

Argument	Description
position	String specifying the position under the current parent in which to insert the new item: ♦ First ♦ Last ♦ Before ♦ After
item	Integer containing an index relative to the current position. The function uses this argument if you specify Before or After for *position*

Return value

Integer. Returns the index of the new item if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the u_lvs pfc_InsertItem event:

```
...
If isvalid(inv_datasource) then
   Return inv_datasource.of_insertitem(ads_source,  &
      al_row, llvi_new, as_position, ai_index)
End If
```

of_IsConfirmOnDelete

Description

Reports whether deletions are confirmed.

Access

Public

Syntax

lvcontrol.instancename.**of_IsConfirmOnDelete** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

Boolean. Returns TRUE if deletions are confirmed and FALSE if they are not.

Examples

This example calls the of_IsConfirmOnDelete function:

```
IF lv_1.inv_datasource.of_IsConfirmOnDelete() THEN
   MessageBox("ListView", "Confirm enabled")
```

```
ELSE
   MessageBox("ListView", "Confirm disabled")
END IF
```

of_Refresh

Description

Updates the ListView with items from the data source.

Access

Public

Syntax

lvcontrol.instancename.**of_Refresh ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_lvs pfc_Refresh event:

```
...
If inv_datasource.of_refresh() < 1 Then
      Return -1
   End If
...
```

of_Register

Registers a data source for the ListView control. There are six syntaxes:

To	Use
Display data retrieved via a DataWindow object	Syntax 1
Display data retrieved via a SQL statement	Syntax 2
Display data from a DataWindow control	Syntax 3
Display data from a DataStore	Syntax 4
Display passed data	Syntax 5
Display data from a file	Syntax 6

Another version of of_Register
There is also an undocumented version of of_Register that is for internal use.

Syntax 1 **To display data retrieved via a DataWindow object**

Description

Establishes a ListView data source by retrieving rows using the specified DataWindow object.

Access

Public

Syntax

lvcontrol.instancename.**of_Register** (*labelcolumn, dwobject* {, *trans* })

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
labelcolumn	String specifying the *dwobject* column that u_lvs uses for ListView item labels
dwobject	String specifying the DataWindow object used to retrieve data
trans (optional)	N_tr Transaction object used to retrieve rows for *dwobject* If you do not specify this argument, *dwobject* must have been saved with the data

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** *dwobject* had no key columns assigned
- **-3** *dwobject* was not provided
- **-4** A specified key column did not exist
- **-5** SetTransObject failed
- **-6** Caching error

Usage

Call this function before calling other functions that customize ListView display.

Examples

This example calls the of_Register function:

```
this.of_SetDataSource(TRUE)
this.inv_datasource.of_Register("emp_lname",  &
   "d_emplist", SQLCA)
this.inv_datasource.of_SetConfirmOnDelete(TRUE)
this.Event pfc_Populate()
```

Syntax 2 **To display data retrieved via a SQL statement**

Description

Establishes a ListView data source by retrieving rows from the database using the specified SQL statement.

Access

Public

Syntax

lvcontrol.instancename.**of_Register** (*labelcolumn, trans, sql*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
labelcolumn	String specifying the column that u_lvs uses for ListView item labels
trans	N_tr Transaction object used to retrieve rows for *sql*
sql	String specifying the SQL SELECT statement used to create the ListView data source

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-4** A specified key column did not exist
- **-5** SetTransObject failed
- **-6** Caching error

Usage

Call this function before calling other functions that customize ListView display.

Examples

This example calls the of_Register function:

```
String  ls_sql
```

```
this.of_SetDataSource(TRUE)
ls_sql = "Select * from employee;"
this.inv_datasource.of_Register("emp_lname",    &
   SQLCA, ls_sql)
this.Event pfc_Populate()
```

Syntax 3 **To display data from a DataWindow control**

Description

Establishes a ListView data source using rows contained in the passed DataWindow control.

Access

Public

Syntax

lvcontrol.instancename.**of_Register** (*labelcolumn*, *dwcontrol*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
labelcolumn	String specifying the column that u_lvs uses for ListView item labels
dwcontrol	DataWindow control containing data to use in the ListView

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** The DataWindow object associated with *dwcontrol* had no key columns assigned
- **-3** *Dwcontrol* has no associated DataWindow object
- **-4** A specified key column did not exist
- **-5** SetTransObject failed
- **-6** Caching error

Usage

Call this function before calling other functions that customize ListView display.

Examples

This example calls the of_Register function:

```
String  ls_sql
```

```
ls_key[1] = "emp_id"
lv_1.inv_datasource.of_Register("emp_lname", dw_1)
lv_1.Event pfc_Populate()
```

Syntax 4 **To display data from a DataStore**

Description

Establishes a ListView data source using rows in the passed DataStore.

Access

Public

Syntax

lvcontrol.instancename.**of_Register** (*labelcolumn*, *datastore*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
labelcolumn	String specifying the column that u_lvs uses for ListView item labels
datastore	DataStore containing data to use in the ListView

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** The DataWindow object associated with *datastore* had no key columns assigned
- **-3** *Datastore* has no associated DataWindow object
- **-4** A specified key column did not exist
- **-5** SetTransObject failed
- **-6** Caching error

Usage

Call this function before calling other functions that customize ListView display.

Examples

This example calls the of_Register function:

```
String  ls_sql

lv_1.inv_datasource.of_Register("emp_lname", ids_data)
lv_1.Event pfc_Populate()
```

Syntax 5 **To display passed data**

Description

Establishes a ListView data source using passed data.

Access

Public

Syntax

lvcontrol.instancename.**of_Register** (*labelcolumn*, *dwobject*, *data*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
labelcolumn	String specifying the column that u_lvs uses for ListView item labels
dwobject	String specifying the DataWindow object whose columns correspond to *data*
data	PowerObject array containing the data to display in the ListView

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-4** A specified key column did not exist
- **-6** Caching error

Usage

Call this function before calling other functions that customize ListView display.

Elements must match
Errors will occur if the elements of the *data* structure do not exactly match the columns of *dwobject*.

The of_Refresh function does not affect data cached with this syntax.

Examples

This example calls the of_Register function:

```
Integer  li_rc, li_cnt
Long     ll_rowcount
// Object structure to contain data
os_powersource  lstr_powersource[ ]
```

```
ll_rowcount = dw_products.rowcount()
for li_cnt = 1 to ll_rowcount
   lstr_powersource[li_cnt] =  &
     dw_products.object.data[li_cnt]
end for
li_rc =  &
   lv_1.inv_datasource.of_Register("description",  &
    "d_products", lstr_powersource)
If li_rc > 0 Then
   li_rc =  &
    lv_1.inv_datasource.of_SetPictureColumn  &
      ("picture_name")
End IF
lv_1.event pfc_populate()
```

Syntax 6 **To display data from a file**

Description Establishes a ListView data source using data from a file.

Access Public

Syntax *lvcontrol.instancename*.**of_Register** (*labelcolumn, dwobject, file*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
labelcolumn	String specifying the column that u_lvs uses for ListView item labels
dwobject	String specifying the DataWindow object whose columns correspond to the data in *file*
file	String specifying the file to be used as the ListView data source

Return value Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-4** A specified key column did not exist
- **-6** Caching error

Usage

Call this function before calling other functions that customize ListView display.

Elements must match
Errors will occur if the elements of the lines in *filename* do not exactly match the columns of *dwobject*.

The of_Refresh function does not affect data cached with this syntax.

Examples

This example calls the of_Register function:

```
Integer  li_rc

li_rc =  &
  lv_1.inv_datasource.of_Register("description",  &
    "d_products", "listview.txt")
If li_rc > 0 Then
   li_rc =  &
   lv_1.inv_datasource.of_SetPictureColumn  &
      ("picture_name")
End IF
lv_1.event pfc_populate()
```

of_RegisterDataSource

Description

Registers the ListView data source with n_cst_dwcache, the PFC caching service.

Access

Protected

Syntax

lvcontrol.instancename.**of_RegisterDataSource** (*method, dwobject, trans, sql, data, dwcontrol, datastore, file*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Argument	Description
method	String specifying the data source population method. Values are: Retrieve DataObjectData PowerObject DataWindowControl DataStoreControl ImportFile SQL
dwobject	String specifying the DataWindow object used by the data source
trans	N_tr specifying the Transaction object used by the data source
sql	String specifying the SQL statement used as the data source
data	PowerObject array containing data used as the data source
dwcontrol	DataWindow control whose associated DataWindow object and data is used as the data source
datastore	DataStore whose associated DataWindow object and data is used as the data source
file	String specifying the file used as the data source

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Register function:

```
...
li_rc = of_registerdatasource(as_method,  &
  as_DWobjectname, atr_obj, &
    as_sql, apo_data, adw_control, ads_control,  &
      as_importfile)
If li_rc < 1 Then
   Return -6
End If
...
```

of_RegisterReportColumn

Description

Adds one or all visible columns to the list of columns displayed in detail view.

Access

Public

Syntax

lvcontrol.instancename.**of_RegisterReportColumn ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

Integer. Returns the number of columns added if the function succeeds and -1 if an error occurs.

Usage

Call this function after calling of_Register.

Examples

This example calls the of_RegisterReportColumn function:

```
Integer li_count

this.of_SetDataSource(TRUE)
this.inv_datasource.of_Register("emp_lname",  &
   "d_emplist", SQLCA)
li_count =  &
   this.inv_datasource.of_RegisterReportColumn()
MessageBox("ListView", String(li_count)  &
   + " columns added")
this.Event pfc_Populate()
```

of_Reset

Description

Clears all data from the data source.

Access

Public

Syntax

lvcontrol.instancename.**of_Reset ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the of_ResetList function:

```
...
this.of_reset()
this.of_clearundo()
...
```

of_ResetList

Description: Clears data from the ListView and the associated data source.

Access: Public

Syntax: *lvcontrol.instancename*.**of_ResetList ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples: This example calls the of_ResetList function:

```
lv_1.inv_datasource.of_ResetList()
```

of_ResetUpdate

Description: Resets the update flags in the DataStore that contains ListView data.

Access: Public

Syntax: *lvcontrol.instancename*.**of_ResetUpdate ( )**

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples: This example calls the of_ResetUpdate function:

```
lv_1.inv_datasource.of_ResetUpdate()
```

of_Retrieve

Description

Retrieves ListView data from rows in the DataStore.

Access

Public

Syntax

lvcontrol.instancename.**of_Retrieve** (*arguments*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
arguments	Twenty-element Any array containing retrieval arguments

Return value

Integer. Returns the number of items retrieved if the function succeeds and -1 if an error occurs.

Usage

Specify retrieval arguments by extending the u_lvs pfc_Retrieve event.

Examples

This example is from the u_lvs of_Retrieve function:

```
...
If isvalid(inv_datasource) then
   ll_rows = inv_datasource.of_retrieve(la_args)
End If
...
```

of_SetCache

Description

Enables or disables n_cst_dwcache.

Access

Protected

Syntax

lvcontrol.instancename.**of_SetCache** (*boolean*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
boolean	Boolean indicating whether to enable (TRUE) or disable (FALSE) n_cst_dwcache

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the Constructor event:

```
of_SetCache(True)
```

of_SetConfirmOnDelete

Description

Specifies whether n_cst_lvsrv_datasource displays a confirmation dialog box when users delete items.

Access

Public

Syntax

lvcontrol.instancename.**of_SetConfirmOnDelete** (*boolean*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
boolean	Boolean specifying whether to display (TRUE) or not display (FALSE) a deletion confirmation dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetConfirmOnDelete function:

```
...
this.of_SetDataSource(TRUE)
ls_key[1] = "emp_id"
this.inv_datasource.of_Register("emp_lname",  &
   ls_key, "d_emplist", SQLCA)
this.inv_datasource.of_SetConfirmOnDelete(FALSE)
...
```

of_SetItemAttributes

Description

Sets ListView item default properties using data from the data source.

Access

Public

Syntax

lvcontrol.instancename.**of_SetItemAttributes** (*datastore, row, item*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
datastore	N_ds containing the DataStore used to create ListView item properties
row	Long specifying the row in *datastore* used to create ListView item properties
item	ListViewItem whose properties are to be set (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_lvs pfc_SetItemAttributes event:

```
If isvalid(inv_datasource) then
   inv_datasource.of_SetItemAttributes(ads_obj, &
      al_row, alvi_item)
End If
...
```

of_SetOverlayPictureColumn

Description

Specifies the name of the DataWindow object column that references the overlay picture.

Access

Public

Syntax

lvcontrol.instancename.**of_SetOverlayPictureColumn** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
column	String specifying one of the following: ♦ The DataWindow object column that u_lvs uses for overlay pictures ♦ A number, which causes u_lvs to use the ListView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetOverlayPictureColumn function:

```
lv_1.inv_datasource.of_SetOverlayPictureColumn("10")
```

of_SetPictureColumn

Description

Specifies the default picture for large icon and small icon view.

Access

Public

Syntax

lvcontrol.instancename.**of_SetPictureColumn** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
column	String specifying one of the following: ♦ The DataWindow object column that u_lvs uses for pictures. This column can contain either a filename or a number, which causes u_lvs to use the ListView's picture index ♦ A number, which causes u_lvs to use the ListView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetPictureColumn function:

```
...
this.inv_datasource.of_Register("emp_lname",  &
   ls_key, "d_emplist", SQLCA)
this.inv_datasource.of_SetPictureColumn("1")
...
```

of_SetStatePictureColumn

Description

Specifies the name of the DataWindow object column that references the state picture.

Access

Public

Syntax

lvcontrol.instancename.**of_SetStatePictureColumn** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
column	String specifying one of the following: ♦ The DataWindow object column that u_lvs uses for state pictures. This column can contain either a filename or a number, which causes u_lvs to use the ListView's picture index ♦ A number, which causes u_lvs to use the ListView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetStatePictureColumn function:

```
...
this.inv_datasource.of_Register("emp_lname",  &
   ls_key, "d_emplist", SQLCA)
this.inv_datasource.of_SetPictureColumn("1")
this.inv_datasource.of_SetStatePictureColumn  &
   ("1")
...
```

of_SetXPosColumn

Description

Specifies the DataWindow object column that u_lvs uses for the item's x coordinate.

Access

Public

Syntax

lvcontrol.instancename.**of_SetXPosColumn** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
column	String specifying the DataWindow object column that specifies the item's x coordinate

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function. This specification applies to large icon and small icon views only.

Examples

This example calls the of_SetXPosColumn function:

```
lv_1.inv_datasource.of_SetXPosColumn("emp_xpos")
```

of_SetYPosColumn

Description

Specifies the DataWindow object column that u_lvs uses for the item's y coordinate.

Access

Public

Syntax

lvcontrol.instancename.**of_SetYPosColumn** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
column	String specifying the DataWindow object column that specifies the item's y coordinate

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function. This specification applies to large icon and small icon views only.

Examples

This example calls the of_SetYPosColumn function:

```
lv_1.inv_datasource.of_SetYPosColumn("emp_ypos")
```

of_UnRegisterReportColumn

Description

Removes the specified column or all columns from the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_UnRegisterReportColumn** ({ *column* })

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
column	Integer specifying the column index (the default is all columns)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If you call this function for the first column, call of_Refresh to refresh the ListView.

Examples

This example calls the of_UnRegisterReportColumn function:

```
...
lv_1.of_UnRegisterReportColumn(2)
...
```

of_Update

Description

Saves all rows in the DataStore associated with the ListView.

Access

Public

Syntax

lvcontrol.instancename.**of_Update** (*accept, resetflag*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
accept	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database
resetflag	Boolean indicating whether the Update function resets the update flags

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Update function:

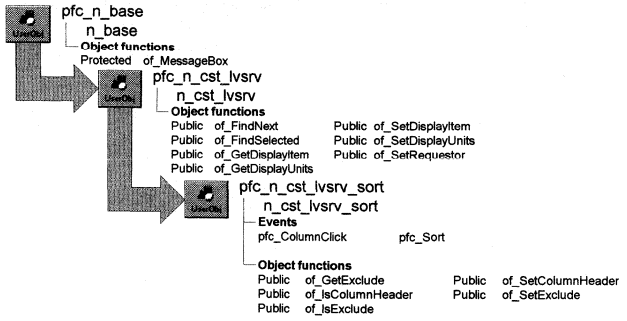
```
lv_1.inv_datasource.of_Update(TRUE, TRUE)
```

n_cst_lvsrv_sort

Description

ListView sort service. ListView sorting applies to detail view only.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use this object:

1 Enable the service by calling the u_lvs of_SetSort function:

```
lv_1.of_SetSort(TRUE)
```

2 Specify whether to enable column header sorting by calling the of_SetColumnHeader function:

```
lv_1.inv_sort.of_SetColumnHeader(TRUE)
```

3 (Optional) Specify columns to exclude from sorting by calling the of_SetExclude function:

```
String  ls_exclude[ ]

ls_exclude[1] = "emp_phone_number"
lv_1.inv_sort.of_SetExclude(ls_exclude)
```

See also

n_cst_lvsrv_datasource
u_lvs

Instance variables

Instance variable	Description	Data type	Access	Usage
ib_columnheadersort	Indicates whether column header sort is enabled	Boolean	Protected	Set with of_SetColumnHeader
ie_currentsort	Specifies current sort type	GRSortType	Protected	Internal
ii_currentsortcol	Specifies current sort column	Integer	Protected	Internal
is_excludecolumns[ ]	Columns for which sorting is disabled	String	Protected	Set with of_SetExclude

Events

N_cst_lvsrv_sort includes precoded events:

pfc_ColumnClick
pfc_Sort

pfc_ColumnClick

Description

Sorts items based on the column whose heading was clicked. If the column was already sorted, this event reverses the sort order.

Syntax

lvcontrol.instancename.EVENT **pfc_ColumnClick** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
column	Integer indicating the column that was clicked

Return value

Integer. Returns 1 if the event succeeds, 0 if no action is taken, and -1 if an error occurs.

Usage

This event executes when the user clicks a column heading in detail view.

Examples

This example is from the u_lvs ColumnClick event:

```
...
If isvalid(inv_sort) then
   inv_sort.event pfc_columnclick(column)
```

```
End If
...
```

pfc_Sort

Description

Performs the sort comparison on the column whose heading was clicked.

Syntax

lvcontrol.instancename.EVENT **pfc_Sort** (*index1*, *index2*, *column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
index1	Integer specifying the index of the first item being compared
index2	Integer specifying the index of the second item being compared
column	Integer specifying the number of the column containing the items to be sorted

Return value

Integer. Returns 1 if *index1* is greater than *index2*, 0 if they are equal, and -1 if *index1* is less than *index2*.

Usage

The u_lvs Sort event calls this event.

Examples

This example is from the u_lvs Sort event:

```
Integer  li_rc

If isvalid(inv_sort) then
   li_rc = inv_sort.event pfc_sort(index1,  &
      index2, column)
End If
Return li_rc
```

Functions

N_cst_lvsrv_sort includes precoded functions:

of_GetExclude	of_SetColumnHeader
of_IsColumnHeader	of_SetExclude

of_IsExclude

of_GetExclude

Description — Retrieves the list of excluded columns.

Access — Public

Syntax — *lvcontrol.instancename*.**of_GetExclude** (*columns*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
columns	String array into which the function places the list of excluded columns (passed by reference)

Return value — Integer. Returns the number of elements in the *columns* array if the function succeeds and -1 if an error occurs.

Examples — This example calls the of_GetExclude function:

```
String  ls_cols[ ]
Integer  li_count, li_return

lb_1.Reset()
li_return = lv_1.inv_sort.of_GetExclude(ls_cols)
FOR li_count = 1 to li_return
   lb_1.AddItem(ls_cols[li_count])
NEXT
```

of_IsColumnHeader

Description — Reports whether column header sorting is enabled.

Access — Public

Syntax — *lvcontrol.instancename*.**of_IsColumnHeader** ( )

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control

Argument	Description
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)

Return value

Boolean. Returns TRUE if column header sorting is enabled and FALSE if it is not.

Examples

This example calls the of_IsColumnHeader function:

```
IF lv_1.inv_sort.of_IsColumnHeader() THEN
   MessageBox("ListView", "Sorting enabled")
ELSE
   MessageBox("ListView", "Sorting disabled")
END IF
```

of_IsExclude

Description

Reports whether a column is eligible for sorting.

Access

Public

Syntax

lvcontrol.instancename.**of_IsExclude** (*column*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
column	Integer specifying the column number or string specifying the column name

Return value

Boolean. Returns TRUE if *column* is excluded and FALSE if it is included.

Examples

This example calls the of_IsExclude function:

```
IF lv_1.inv_sort.of_IsExclude("emp_fname") THEN
   MessageBox("ListView", "Excluded")
ELSE
   MessageBox("ListView", "Included")
END IF
```

of_SetColumnHeader

Description

Enables or disables column header sorting.

Access

Public

Syntax

lvcontrol.instancename.**of_SetColumnHeader** (*boolean*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
boolean	Boolean specifying whether to enable (TRUE) or disable (FALSE) column header sorting

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetColumnHeader function:

```
this.of_SetSort(TRUE)
this.inv_sort.of_SetColumnHeader(TRUE)
```

of_SetExclude

Description

Specifies columns to be excluded from sorting.

Access

Public

Syntax

lvcontrol.instancename.**of_SetExclude** (*columns*)

Argument	Description
lvcontrol	Instance name of the u_lvs-based ListView control
instancename	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
columns	String array containing columns to be excluded from sorting

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetExclude function:

```
String  ls_exclude[ ]
this.of_SetSort(TRUE)
this.inv_sort.of_SetColumnHeader(TRUE)
```

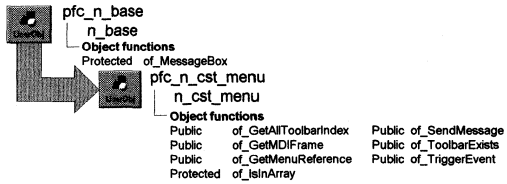
```
ls_exclude[1] = "emp_phone_number"
this.inv_sort.of_SetExclude(ls_exclude)
```

n_cst_menu

Description

Menu service object. The m_master of_SendMessage function, the preference service, and the w_toolbars dialog box use n_cst_menu functions to access menu and frame information.

Ancestry



Library

PFCWNSRV.PBL
PFEWNSRV.PBL

Usage

To use this object:

1 Declare a variable of type n_cst_menu. Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements:

```
n_cst_menu  lnv_menu
```

2 Call n_cst_menu functions as needed (this example is from the m_master of_SendMessage function):

```
...
li_rc = lnv_menu.of_GetMDIframe(this, lw_frame)
IF NOT IsNull (lw_frame) AND &
   IsValid(lw_frame) THEN
      lb_frameexists = TRUE
END IF
...
```

See also

m_master
n_cst_winsrv_preference

Functions

N_cst_menu contains object functions:

of_GetAllToolbarIndex	of_SendMessage
of_GetMDIFrame	of_ToolbarExists

of_GetMenuReference	of_TriggerEvent
of_IsInArray	

of_GetAllToolbarIndex

Description

Retrieves an array containing all unique toolbar item index values.

Access

Public

Syntax

instancename.**of_GetAllToolbarIndex** (*menu, index*)

Argument	Description
instancename	Instance name of n_cst_menu
menu	Menu variable containing the menu whose toolbar index is returned
index	Integer array into which the function places all unique ToolbarItemBarIndex values (passed by reference)

Return value

Integer. Returns the size of the *index* array if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetAllToolbarIndex function:

```
...
IF lnv_menu.of_GetAllToolbarIndex  &
   (iw_requestor.MenuId, li_barindexarray) < 0 THEN
      Return -1
END IF
...
```

of_GetMDIFrame

Description

Retrieves the MDI frame window.

Access

Public

Syntax

instancename.**of_GetMDIFrame** (*menu, frame*)

Argument	Description
instancename	Instance name of n_cst_menu
menu	Menu variable containing the menu for which to find the MDI frame

Argument	Description
frame	Window variable into which the function places the frame window (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetMDIFrame function:

```
Integer  li_rc
Window   lw_obj
n_cst_menu  lnv_menu

lnv_menu.of_GetMDIFrame(iw_requestor.MenuID, lw_obj)
IF IsValid(lw_obj) THEN
   lw_obj.SetRedraw(FALSE)
ELSE
   iw_requestor.SetRedraw(False)
END IF
...
```

of_GetMenuReference

Description

Retrieves a reference to the specified menu item.

Access

Public

Syntax

instancename.**of_GetMenuReference** (*menu, itemname, itemref*)

Argument	Description
instancename	Instance name of n_cst_menu
menu	Menu variable containing the menu for which to retrieve the menu item reference
itemname	String specifying the menu item name
itemref	Menu variable into which the function places a reference to the menu item (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, 0 if *itemname* is not found, and -1 if an error occurs.

Usage

Use the retrieved reference to control menu item properties.

Examples

This example calls the of_GetMenuReference function:

```
n_cst_menu  lnv_menu
Menu  lm_item, lm_menu

lm_menu = parent.MenuID
lnv_menu.of_GetMenuReference  &
   (lm_menu, "m_save", lm_item)
lm_item.Enabled = FALSE
```

of_IsInArray

Description

Reports whether the specified ToolbarItemBarIndex is in the array.

Access

Protected

Syntax

instancename.**of_IsInArray** (*array*, *index*)

Argument	Description
instancename	Instance name of n_cst_menu
array	Integer array containing currently stored ToolbarItemBarIndex numbers
index	Integer specifying the ToolbarItemBarIndex to be tested

Return value

Boolean. Returns TRUE if *index* is in *array* and FALSE if it is not.

Usage

Internal.

Examples

This example is from the of_GetAllToolbarIndex function.

```
...
IF Len(am_source.toolbaritemname) > 0 THEN
   IF NOT of_IsInArray(ai_barindex, &
     am_source.ToolbarItemBarIndex) THEN
       ai_barindex[UpperBound(ai_barindex)+1] &
         = am_source.ToolbarItemBarIndex
   END IF
END IF
...
```

of_SendMessage

Description

Sends the passed message to the current window through the pfc_MessageRouter event.

Access

Public

Syntax

instancename.**of_SendMessage** (*message*)

Argument	Description
instancename	Instance name of n_cst_menu
message	String specifying the user event to be triggered by the pfc_MessageRouter event

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

PFC menu items call this function to trigger events on the associated window and window controls.

Examples

This example calls the of_SendMessage function:

```
n_cst_menu  lnv_menu

lnv_menu.of_SendMessage("pfc_Open")
```

of_ToolbarExists

Description

Reports whether the passed menu has a toolbar.

Access

Public

Syntax

instancename.**of_ToolbarExists** (*menu*)

Argument	Description
instancename	Instance name of n_cst_menu
menu	Menu variable containing the menu to be tested for the existence of a toolbar

Return value

Boolean. Returns TRUE if *menu* has a toolbar and FALSE if it does not.

Examples

This example is from the n_cst_preference of_Save function:

```
...
IF NOT lnv_menu.of_ToolBarExists &
   (iw_requestor.MenuId) THEN
   Return 1
END IF
...
```

of_TriggerEvent

Description

Triggers the specified user event on the sheet associated with the passed menu.

Access

Public

Syntax

instancename.**of_TriggerEvent** (*menu*, *event*)

Argument	Description
instancename	Instance name of n_cst_menu
menu	Menu variable containing the menu for which to trigger the event
event	String specifying the user event to trigger

Return value

Integer. Returns 1 if the function succeeds and -1 if *event* does not exist has or no script.

Examples

This example calls the of_TriggerEvent function:

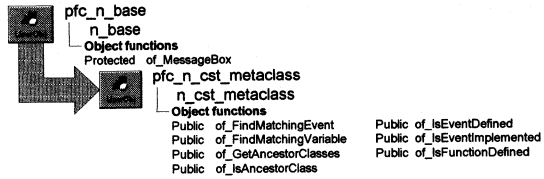
```
n_cst_menu  lnv_menu
Integer     li_return

li_return = lnv_menu.of_TriggerEvent &
   (parent.menuid, "ue_options")
IF li_return = -1 THEN
   MessageBox("Menu", "Options not enabled")
END IF
```

n_cst_metaclass

Description

Metaclass service. This object contains functions that provide information on the functions, events, and variables defined within another object.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use this object:

1 Create a variable of type n_cst_metaclass.

Because PFC defines this object with the autoinstantiate option, there is no need to code CREATE or DESTROY statements.

2 Call n_cst_metaclass functions as needed:

```
boolean  lb_ancestor
n_cst_metaclass  lnv_metaclass
n_cst_conversion  lnv_conversion
classdefinition  lcd_obj

lcd_obj = findClassDefinition ("w_sheet")
lb_ancestor = lnv_metaclass.of_isAncestorClass  &
   (lcd_obj, "pfc_w_master")
if isNull (lb_ancestor) then
   messagebox ("MetaClass", "class not found")
else
   messagebox ("MetaClass",  &
      lnv_conversion.of_string (lb_ancestor))
end if
```

See also

n_cst_luw

Functions

N_cst_metaclass includes precoded functions:

of_FindMatchingEvent	of_IsEventDefined
of_FindMatchingVariable	of_IsEventImplemented
of_GetAncestorClasses	of_IsFunctionDefined
of_IsAncestorClass	

of_FindMatchingEvent

Description

Retrieves the script definition for a given object and event.

Access

Public

Syntax

instancename.**of_FindMatchingEvent** (*object*, *event*)

Argument	Description
instancename	Instance name of n_cst_metaclass
object	ClassDefinition specifying the object containing *event*
event	String specifying the event name

Return value

ScriptDefinition. Returns the script definition if the function succeeds and NULL if *event* cannot be found.

Examples

This example calls the of_FindMatchingEvent function:

```
classdefinition  lcd_object
scriptdefinition  lscrd_object
n_cst_metaclass  lnv_metaclass

lcd_object = findClassDefinition ("pfc_u_dw")
lscrd_object = lnv_metaclass.of_findMatchingEvent  &
   (lcd_object, "pfc_update")
if isNull (lscrd_object) or not isValid  &
   (lscrd_object) then
     messagebox ("MetaClass",  &
       "Bad class or can't find event definition")
else
   messagebox ("MetaClass", lscrd_object.name)
end if
```

of_FindMatchingVariable

Description

Retrieves the variable definition for a given object or script and variable.

Access

Public

Syntax

instancename.**of_FindMatchingVariable** (*classorscript*, *variable*)

Argument	Description
instancename	Instance name of n_cst_metaclass
classorscript	ClassDefinition of the object containing *variable* or ScriptDefinition of the script containing *variable*
variable	String specifying the variable name

Return value

VariableDefinition. Returns the variable definition if the function succeeds and NULL if an error occurs.

Usage

If you specify a ScriptDefinition as the first argument, the function looks for *variable* in both local variables and arguments.

Examples

This example calls the of_FindMatchingVariable function:

```
string  ls_argtype[ ]
classdefinition  lcd_object
variabledefinition  lvrd_object
scriptdefinition  lscrd_object
n_cst_metaclass  lnv_metaclass

lcd_object = findClassDefinition("pfc_u_dw")
ls_argtype[1] = "Boolean"
ls_argtype[2] = "Boolean"
lscrd_object = lcd_object.findMatchingFunction  &
   ("of_Update", ls_argtype[ ])
if isNull (lscrd_object) or not isValid  &
   (lscrd_object) then
     messagebox ("", "can't find script definition")
end if
lvrd_object = lnv_metaclass.of_findMatchingVariable &
   (lscrd_object, "li_rc")
if not isValid (lvrd_object) or  &
  isNull (lvrd_object) then
     messagebox ("MetaClass",  &
      "scriptdefinition not valid or var not found")
else
```

```
    messagebox ("MetaClass", lvrd_object.name)
end if
```

of_GetAncestorClasses

Description

Retrieves all ancestor classes for a specified class.

Access

Public

Syntax

instancename.**of_GetAncestorClasses** (*sourceclass*, *ancestorclass*)

Argument	Description
instancename	Instance name of n_cst_metaclass
sourceclass	String or ClassDefinition specifying the class for which to retrieve ancestor classes
ancestorclass	ClassDefinition array into which the function places references to ancestor classes (passed by reference)

Return value

Integer. Returns the number of elements in the *ancestorclass* array if the function succeeds and -1 if an error occurs.

Usage

The elements in *ancestorclass* are ordered in ascending order (that is, element 1 contains the immediate ancestor and the upperbound contains the highest level ancestor).

Examples

This example calls the of_GetAncestorClasses function:

```
n_cst_metaclass  lnv_metaclass
classdefinition  lcd_ancestor[ ]
integer  li_ubound, li_cnt

lnv_metaclass.of_GetAncestorClasses  &
   ("w_sheet", lcd_ancestor[ ])
li_ubound = UpperBound (lcd_ancestor[ ])
lb_1.reset()
for li_cnt = 1 to li_ubound
   lb_1.addItem (lcd_ancestor[li_cnt].name)
next
```

of_IsAncestorClass

Description

Reports whether a specified class is an ancestor of another specified class.

Access

Public

Syntax

instancename.**of_IsAncestorClass** (*sourceclass*, *ancestorclass*)

Argument	Description
instancename	Instance name of n_cst_metaclass
sourceclass	String or ClassDefinition containing the possible descendent class
ancestorclass	String containing the class that might be an ancestor of *sourceclass*

Return value

Boolean. Returns TRUE if *sourceclass* is a descendant of *ancestorclass* and FALSE if it is not.

Examples

This example calls the of_IsAncestorClass function:

```
boolean  lb_ancestor
n_cst_metaclass  lnv_metaclass
n_cst_conversion  lnv_conversion
classdefinition  lcd_obj

lcd_obj = findClassDefinition ("w_sheet")
lb_ancestor = lnv_metaclass.of_isAncestorClass  &
   (lcd_obj, "pfc_w_master")
if isNull (lb_ancestor) then
   messagebox ("MetaClass", "class not found")
else
   messagebox ("MetaClass",  &
      lnv_conversion.of_string (lb_ancestor))
end if
```

of_IsEventDefined

Description

Reports whether the specified class contains the specified event.

Access

Public

Syntax

instancename.**of_IsEventDefined** (*class*, *event*)

Argument	Description
instancename	Instance name of n_cst_metaclass
class	String or ClassDefinition to be tested

Argument	Description
event	String specifying the event

Return value

Boolean. Returns TRUE if *class* contains *event* and FALSE if it does not.

Examples

This example calls the of_IsEventDefined function:

```
boolean  lb_defined
n_cst_metaclass  lnv_metaclass
n_cst_conversion lnv_conversion
classdefinition  lcd_obj

lcd_obj = findClassDefinition ("pfc_n_ds")
lb_defined = lnv_metaclass.of_isEventDefined  &
   (lcd_obj, "of_accepttext")
if isNull (lb_defined) then
   messageBox ("MetaClass", "Class not found")
else
   messageBox ("MetaClass",   &
      lnv_conversion.of_string (lb_defined))
end if
```

of_IsEventImplemented

Description

Reports whether the specified class implements the specified event.

Access

Public

Syntax

instancename.**of_IsEventImplemented** (*class*, *event*)

Argument	Description
instancename	Instance name of n_cst_metaclass
class	String or ClassDefinition to be tested
event	String specifying the event

Return value

Boolean. Returns TRUE if *class* implements *event* and FALSE if it does not.

Examples

This example calls the of_IsEventImplemented function:

```
boolean  lb_defined
n_cst_metaclass  lnv_metaclass
n_cst_conversion  lnv_conversion
```

```
classdefinition  lcd_obj

lcd_obj = findClassDefinition ("pfc_n_ds")
lb_defined = lnv_metaclass.of_isEventImplemented  &
   (lcd_obj, "destructor")
if isNull (lb_defined) then
   messageBox ("MetaClass", "class not found")
else
   messageBox ("MetaClass",  "Implemented = "  &
      + lnv_conversion.of_string (lb_defined))
end if
```

of_IsFunctionDefined

Description — Reports whether the specified class contains the specified function.

Access — Public

Syntax — *instancename*.**of_IsFunctionDefined** (*class*, *function*)

Argument	Description
instancename	Instance name of n_cst_metaclass
class	String or ClassDefinition to be tested
function	String specifying the function

Return value — Boolean. Returns TRUE if *class* contains *function* and FALSE if it does not.

Examples — This example is from the n_cst_luw of_AcceptText function:

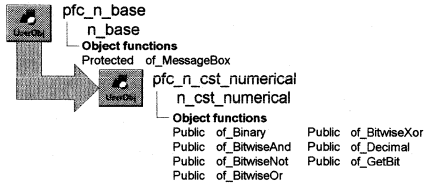
```
...
lb_defined = inv_metaclass.of_isFunctionDefined &
   (lpo_tocheck.ClassDefinition,  &
      "of_AcceptText", ls_args)
...
```

n_cst_numerical

Description

Binary number service. This service provides functions that you can call to access binary data. For example, you can call the of_GetBit function to determine if a specified bit is on or off.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use numerical functions:

1 Declare a variable of type n_cst_numerical:

```
n_cst_numerical  inv_numerical
```

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements.

2 Call numerical service functions as needed.

This example assumes an inv_numerical instance variable:

```
Long ll_base10
String ls_binary

ll_base10 = Long(sle_base10.text)
ls_binary = inv_numerical.of_Binary(ll_base10)

MessageBox("Numerical", &
String(ll_base10) + " base 10 is equal to " + &
   ls_binary + " in binary.")
```

See also

n_cst_conversion
n_cst_string

Functions

N_cst_numerical includes precoded object functions:

of_Binary	of_BitwiseXor
of_BitwiseAnd	of_Decimal
of_BitwiseNot	of_GetBit
of_BitwiseOr	

of_Binary

Description

Determines the binary representation of a positive base 10 number.

Access

Public

Syntax

instancename.**of_Binary** (*number*)

Argument	Description
instancename	Instance name of n_cst_numerical
number	Long to be converted to binary. This number must be positive

Return value

String. Retrieves the binary representation of *number*. If *number* is invalid or NULL, the function returns NULL.

Examples

This example calls the of_Binary function. It assumes an inv_numerical instance variable:

```
Long  ll_base10
String  ls_binary

ll_base10 = Long(sle_base10.text)
ls_binary = inv_numerical.of_Binary(ll_base10)

MessageBox("Numerical", &
   String(ll_base10) + " base 10 is equal to " + &
      ls_binary + " in binary.")
```

of_BitwiseAnd

Description

Performs a bitwise AND operation on each bit of two passed values. In a bitwise AND operation:

1 AND 0 evaluates to 0
0 AND 0 evaluates to 0

1 AND 1 evaluates to 1

Access

Public

Syntax

instancename.**of_BitwiseAnd** (*value1*, *value2*)

Argument	Description
instancename	Instance name of n_cst_numerical
value1	Long containing one of the values used in the bitwise AND operation
value2	Long containing the other value used in the bitwise AND operation

Return value

Long. Returns the result of the AND operation on each of *value1*'s and *value2*'s bits.

Examples

This example is from the n_cst_filesrv of_IncludeFile function:

```
n_cst_numerical   lnv_numeric
...
IF lnv_numeric.of_BitwiseAnd &
   (aul_FileAttrib, al_AttribMask) > 0 THEN
   Return TRUE
END IF
```

of_BitwiseNot

Description

Performs a bitwise NOT operation on each bit of a passed value. In a bitwise NOT operation:

1 evaluates to 0
0 evaluates to 1

Access

Public

Syntax

instancename.**of_BitwiseNot** (*value*)

Argument	Description
instancename	Instance name of n_cst_numerical
value	Long containing one of the value used in the bitwise NOT operation

Return value

Long. Returns the result of the NOT operation on each of *value*'s bits.

Examples

This example calls the of_BitwiseNot function. It assumes an inv_numerical instance variable:

```
Long   ll_original, ll_not

ll_original = Long(sle_not.text)
ll_not = inv_numerical.of_BitwiseNot &
   (ll_original)

MessageBox("Numerical", "Binary original is " &
   + String(ll_original) + ". Bitwise NOT is " &
   + String(ll_not) )
```

of_BitwiseOr

Description

Performs a bitwise OR operation on each bit of two passed values. In a bitwise OR operation:

1 OR 0 evaluates to 1
0 OR 0 evaluates to 0
1 OR 1 evaluates to 1

Access

Public

Syntax

instancename.**of_BitwiseOr** (*value1*, *value2*)

Argument	Description
instancename	Instance name of n_cst_numerical
value1	Long containing one of the values used in the bitwise OR operation
value2	Long containing the other value used in the bitwise OR operation

Return value

Long. Returns the result of the OR operation on each of *value1*'s and *value2*'s bits.

Examples

This example calls the of_BitwiseOr function. It assumes an inv_numerical instance variable:

```
Long   ll_value1, ll_value2, ll_result

ll_value1 = Long(sle_value1.text)
ll_value2 = Long(sle_value2.text)
ll_result = inv_numerical.of_BitwiseOr &
   (ll_value1, ll_value2)
```

```
MessageBox("Numerical", "Value 1 is " &
    + sle_value1.text + " Value 2 is " &
    + sle_value2.text + " Bitwise OR is " &
    + String(ll_result) )
```

of_BitwiseXor

Description

Performs a bitwise exclusive OR (XOR) operation on each bit of two passed values.

In a bitwise XOR operation:

1 XOR 0 evaluates to 1
0 XOR 0 evaluates to 0
1 XOR 1 evaluates to 0

Access

Public

Syntax

instancename.**of_BitwiseXor** (*value1*, *value2*)

Argument	Description
instancename	Instance name of n_cst_numerical
value1	Long containing one of the values used in the bitwise XOR operation
value2	Long containing the other value used in the bitwise XOR operation

Return value

Long. Returns the result of the XOR operation on each of *value1's* and *value2*'s bits.

Examples

This example calls the of_BitwiseXor function. It assumes an inv_numerical instance variable:

```
Long   ll_value1, ll_value2, ll_result

ll_value1 = Long(sle_value1.text)
ll_value2 = Long(sle_value2.text)
ll_result = inv_numerical.of_BitwiseXor &
   (ll_value1, ll_value2)

MessageBox("Numerical", "Value 1 is " &
   + sle_value1.text + " Value 2 is " &
   + sle_value2.text + " Bitwise XOR is " &
```

```
    + String(ll_result) )
```

of_Decimal

Description

Retrieves the base 10 representation of a binary number.

Access

Public

Syntax

instancename.**of_Decimal** (*binaryvalue*)

Argument	Description
instancename	Instance name of n_cst_numerical
binaryvalue	String containing the binary value whose positive base 10 value is returned

Return value

Long. Returns the positive base 10 representation of *binaryvalue*. If *binaryvalue* is invalid, the function returns -1; if *binaryvalue* is NULL, the function returns NULL.

Examples

This example calls the of_Decimal function. It assumes an inv_numerical instance variable:

```
Long   ll_base10

ll_base10 = inv_numerical.of_Decimal &
   (sle_binary.text)
MessageBox("Numerical", &
   sle_binary.text + " in binary is equal to " + &
      String(ll_base10) + " in Base 10.")
```

of_GetBit

Description

Reports whether the specified binary bit of a passed base 10 argument is on (1) or off (0).

Access

Public

Syntax

instancename.**of_GetBit** (*value*, *bit*)

Argument	Description
instancename	Instance name of n_cst_numerical
value	Long whose on/off value is to be determined

Argument	Description
bit	UnsignedInteger indicating the bit position in *value*, from right to left

Return value

Boolean. Returns TRUE if the bit is on and FALSE if it is off.

Usage

The Windows Software Development Kit (SDK) includes many functions that return bit values. This function can help you access these bit values.

Examples

This example is from the n_cst_filesrvwin16 of_DirList function:

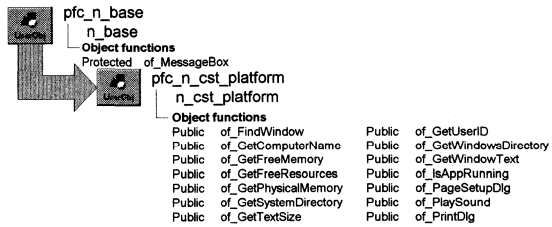
```
n_cst_numerical   lnv_numeric
Integer   li_entries
Uint  lui_attrib
...
// Set file attributes
anv_dirlist[li_entries].ib_readonly = &
   lnv_numeric.of_GetBit(lui_attrib, 1)
...
```

n_cst_platform

Description

Ancestor object for platform-specific functions. Most of the functions in this object are virtual functions and return -1, FALSE, or an empty string. A **virtual function** is an empty placeholder function; the actual processing occurs in its descendants. Defining virtual functions in an ancestor object is a technique that allows you to define a reference variable with the ancestor's data type but instantiate the variable with the descendant's data type. PFC contains descendent objects for each supported platform (for example, 16-bit windows, 32-bit windows, Macintosh, and UNIX).

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.BPL

Object relationships

s_pagesetupattrib
s_printdlgattrib
w_pagesetup

Usage

To use the platform service:

1 Instantiate the appropriate platform services object by calling the f_SetPlatform global function. This function examines your environment and instantiates the appropriate user object.

2 Call platform services functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_platform) function, which returns 0, -1, FALSE, or an empty string.

Descendants

n_cst_platformaix (see n_cst_platformsol2)
n_cst_platformhpux (see n_cst_platformsol2)
n_cst_platformmac
n_cst_platformsol2
n_cst_platformwin16
n_cst_platformwin32

See also

f_SetPlatform

n_cst_filesrv

Instance variables

N_cst_platform contains an instance variable:

Instance variable	Description	Data type	Access	Usage
is_separator	Separator character	String	Protected	Used by the operating system when specifying fully qualified pathnames (default is backslash)

Functions

N_cst_platform includes precoded object functions, which are virtual functions with corresponding versions in n_cst_platform descendants:

of_FindWindow	of_GetUserID
of_GetComputerName	of_GetWindowsDirectory
of_GetFreeMemory	of_GetWindowText
of_GetFreeResources	of_IsAppRunning
of_GetPhysicalMemory	of_PageSetupDlg
of_GetSystemDirectory	of_PlaySound
of_GetTextSize	of_PrintDlg

By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

of_FindWindow

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_FindWindow** (*windowname*)

Argument	Description
instancename	Instance name of n_cst_platform
windowname	String

Return value: UnsignedInteger. Always returns 0.

See also: n_cst_platformsol2 of_FindWindow function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin16 of_FindWindow function
n_cst_platformwin32 of_FindWindow function

of_GetComputerName

Description: Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access: Public

Syntax: *instancename*.**of_GetComputerName** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value: String. Always returns an empty string.

Usage: This function is not supported on 16-bit Windows.

See also: n_cst_platformmac of_GetComputerName function
n_cst_platformsol2 of_GetComputerName function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin32 of_GetComputerName function

of_GetFreeMemory

Description: Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access: Public

Syntax: *instancename*.**of_GetFreeMemory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

Long. Always returns -1.

See also

n_cst_platformmac of_GetFreeMemory function
n_cst_platformsol2 of_GetFreeMemory function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin16 of_GetFreeMemory function
n_cst_platformwin32 of_GetFreeMemory function

of_GetFreeResources

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetFreeResources** (*type*)

Argument	Description
instancename	Instance name of n_cst_platform
type	Integer

Return value

Integer. Always returns -1.

Usage

This function is not supported on 32-bit Windows, Macintosh, and UNIX.

See also

n_cst_platformwin16 of_GetFreeResources function

of_GetPhysicalMemory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetPhysicalMemory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

Long. Always returns -1.

Usage

This function is not supported on 16-bit Windows.

See also

n_cst_platformmac of_GetPhysicalMemory function
n_cst_platformsol2 of_GetPhysicalMemory function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin32 of_GetPhysicalMemory function

of_GetSystemDirectory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetSystemDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Always returns an empty string.

See also

n_cst_platformmac of_GetSystemDirectory function
n_cst_platformsol2 of_GetSystemDirectory function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin16 of_GetSystemDirectory function
n_cst_platformwin32 of_GetSystemDirectory function

of_GetTextSize

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetTextSize** (*window, text, font, fontsize, bold, italic, underline, height, width*)

Argument	Description
instancename	Instance name of n_cst_platform
window	Window
text	String
font	String
fontsize	Integer
bold	Boolean
italic	Boolean
underline	Boolean
height	Integer (passed by reference)
width	Integer (passed by reference)

Return value

Integer. Always returns -1.

Usage

This function is not supported on Macintosh.

See also

n_cst_platformwin16 of_GetTextSize function
n_cst_platformsol2 of_GetTextSize function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin32 of_GetTextSize function

of_GetUserID

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetUserID** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Always returns an empty string.

See also

n_cst_platformmac of_GetUserID function
n_cst_platformsol2 of_GetUserID function (also in n_cst_platformaix and

n_cst_platformhpux)
n_cst_platformwin16 of_GetUserID function
n_cst_platformwin32 of_GetUserID function

of_GetWindowsDirectory

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetWindowsDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Always returns an empty string.

Usage

This function is not supported on Macintosh.

See also

n_cst_platformwin16 of_GetWindowsDirectory function
n_cst_platformsol2 of_GetWindowsDirectory function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin32 of_GetWindowsDirectory function

of_GetWindowText

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_GetWindowText** (*window*)

Argument	Description
instancename	Instance name of n_cst_platform
window	UnsignedInteger

Return value

String. Always returns an empty string.

Usage

This function is not supported on Macintosh.

See also

n_cst_platformsol2 of_GetWindowText function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin16 of_GetWindowText function
n_cst_platformwin32 of_GetWindowText function

of_IsAppRunning

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_IsAppRunning** (*appname*)

Argument	Description
instancename	Instance name of n_cst_platform
appname	String

Return value

Boolean. Always returns NULL.

Usage

This function is not supported on 32-bit Windows, Macintosh, and UNIX.

See also

n_cst_platformwin16 of_IsAppRunning function

of_PageSetupDlg

Description

Displays the w_pagesetup dialog box, which allows you to control print settings.

Access

Public

Syntax

instancename.**of_PageSetupDlg** (*setupinfo*)

Argument	Description
instancename	Instance name of n_cst_platform
setupinfo	S_pagesetupattrib instance into which the function places page setup information (passed by reference)

Return value

Long. Returns 1 if the function succeeds, 0 if the user canceled out of the dialog box, and -1 if an error occurs.

of_PlaySound

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_PlaySound** (*filename* {, *beeponfailure* })

Argument	Description
instancename	Instance name of n_cst_platform
filename	String
beeponfailure	Boolean

Return value

Integer. Always returns -1.

See also

n_cst_platformmac of_PlaySound function
n_cst_platformsol2 of_PlaySound function (also in n_cst_platformaix and n_cst_platformhpux)
n_cst_platformwin16 of_PlaySound function
n_cst_platformwin32 of_PlaySound function

of_PrintDlg

Description

Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type n_cst_platform for use with all n_cst_platform descendants.

Access

Public

Syntax

instancename.**of_PrintDlg** (*printinfo, window*)

Argument	Description
instancename	Instance name of n_cst_platform
printinfo	S_printdlgattrib instance into which the function places print information (passed by reference)
window	Window

Return value

Integer. Always returns -1.

See also

n_cst_platformwin16 of_PrintDlg function
n_cst_platformwin32 of_PrintDlg function

n_cst_platformmac

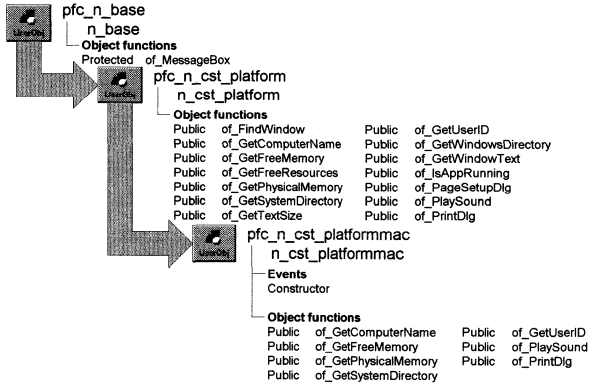
Description

Platform-specific functions for applications running on the Macintosh. Platform-specific functions require a call to a platform-specific external function. When you call the f_SetPlatform function, it detects the operating environment and instantiates the correct platform-specific object.

The functions in this object call platform-specific external functions. To see a complete list of external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar.

This object calls functions in the platform-specific Macintosh module PFCMacPlatformSrv.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use the platform service:

1 Instantiate the appropriate platform services object by calling the f_SetPlatform global function. This function examines your environment and instantiates the appropriate user object.

2 Call platform services functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_platform) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_memorystatus	ul_availpagefile	UnsignedLong
	ul_availphys	UnsignedLong
	ul_availvirtual	UnsignedLong
	ul_length	UnsignedLong
	ul_memoryload	UnsignedLong
	ul_totalpagefile	UnsignedLong
	ul_totalphys	UnsignedLong
	ul_totalvirtual	UnsignedLong
os_size	l_cx	Long
	l_cy	Long

See also

f_SetPlatform
n_cst_filesrv
n_cst_platform
n_cst_platformsol2
n_cst_platformwin16
n_cst_platformwin32

Events

N_cst_platformmac includes a precoded event script for the Constructor event:

Constructor

Constructor

Description

Initializes the is_separator instance variable.

Usage

This event executes when the object is created.

Functions

N_cst_platformmac includes precoded object functions:

of_GetComputerName	of_GetUserID
of_GetFreeMemory	of_PlaySound
of_GetPhysicalMemory	of_PrintDlg
of_GetSystemDirectory	

of_GetComputerName

Description — Retrieves the name of the current workstation.

Access — Public

Syntax — *instancename*.**of_GetComputerName** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value — String. Returns the name of the current workstation.

Usage — This function calls the GetComputerName external function.

Examples — This example calls the of_GetComputerName function:

```
String   ls_computer

ls_computer = inv_platform.of_GetComputerName()
MessageBox("Platform", "Current workstations is: " &
   + ls_computer )
```

of_GetFreeMemory

Description — Retrieves the number of bytes of available memory.

Access — Public

Syntax — *instancename*.**of_GetFreeMemory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value — Long. Returns the number of bytes of available memory.

Usage

This function calls the GetFreeMemory external function.

Examples

This example calls the of_GetFreeMemory function:

```
Long   ll_free

ll_free = inv_platform.of_GetFreeMemory()
MessageBox("Platform", String(ll_free) &
   + " bytes of free memory" )
```

of_GetPhysicalMemory

Description

Retrieves the total number of bytes of physical memory (RAM).

Access

Public

Syntax

instancename.**of_GetPhysicalMemory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value

Long. Returns the number of bytes of memory.

Usage

This function calls the GetPhysicalMemory external function.

Examples

This example calls the of_GetPhysicalMemory function:

```
Long   ll_total

ll_total = inv_platform.of_GetPhysicalMemory( )
MessageBox("Platform", String(ll_total) &
   + " bytes of RAM (total)" )
```

of_GetSystemDirectory

Description

Retrieves the path of the system folder.

Access

Public

Syntax

instancename.**of_GetSystemDirectory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Returns the path of the system folder if the function succeeds and an empty string if it cannot determine the system folder.

Usage

This function calls the GetSystemDirectory external function.

Examples

This example calls the of_GetSystemDirectory function:

```
String   ls_system

ls_system = inv_platform.of_GetSystemDirectory()
IF ls_system <> "" THEN
   MessageBox("Platform", "System folder is: " &
      + String(ls_system) )
ELSE
   MessageBox("Platform", &
      "Unable to determine system folder")
END IF
```

of_GetUserID

Description

Retrieves the current network logon name.

Access

Public

Syntax

instancename.**of_GetUserID** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Returns the current network logon ID if the function succeeds and an empty string if no logon ID can be found.

Usage

This function calls the GetUserID external function.

Examples

This example calls the of_GetUserID function:

```
String   ls_userid

ls_userid = inv_platform.of_GetUserID( )
IF ls_userid <> "" THEN
   MessageBox("Platform", "User ID is: " &
      + String(ls_userid) )
ELSE
   MessageBox("Platform", &
      "Unable to determine user ID")
```

```
END IF
```

of_PlaySound

Description: Plays the specified sound file.

Access: Public

Syntax: *instancename*.**of_PlaySound** (*filename*)

Argument	Description
instancename	Instance name of n_cst_platform
filename	String specifying the complete pathname of the sound file to play

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: This function calls the PlaySound external function.

Examples: This example calls the of_PlaySound function:

```
String   ls_sound

ls_sound=sle_soundfile.text
IF inv_platform.of_PlaySound(ls_sound) = -1 THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      ("No sound capabilities")
END IF
```

of_PrintDlg

Description: Displays an operating-system-specific Print dialog box.

Access: Public

Syntax: *instancename*.**of_PrintDlg** (*printinfo, window*)

Argument	Description
instancename	Instance name of n_cst_platform
printinfo	S_printdlgattrib instance containing print information (passed by reference)
window	Window variable containing the window opening the Print dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs or if the user canceled out of the dialog box.

Examples

This example is from the u_dw pfc_PrintDlg event:

```
...
li_copies = Integer(ls_copies)
astr_printdlg.l_copies = li_copies
ls_collate = this.object.datawindow.print.collate
lb_collate = lnv_conversion.of_Boolean(ls_collate)
astr_printdlg.b_collate = lb_collate
astr_printdlg.l_frompage = 1
astr_printdlg.l_minpage = 1
...
this.Event pfc_PrePrintDlg(astr_printdlg)

f_SetPlatform(lnv_platform, TRUE)
this.of_GetParentWindow(lw_parent)
ll_rc = &
   lnv_platform.of_PrintDlg &
      astr_printdlg, lw_parent)
f_SetPlatform(lnv_platform, FALSE)
...
```

n_cst_platformsol2

Description

Platform-specific functions for applications running under UNIX. Platform-specific functions require a call to a platform-specific external function. When you call the f_SetPlatform function, it detects the operating environment and instantiates the correct platform-specific object.

HP/UX and AIX

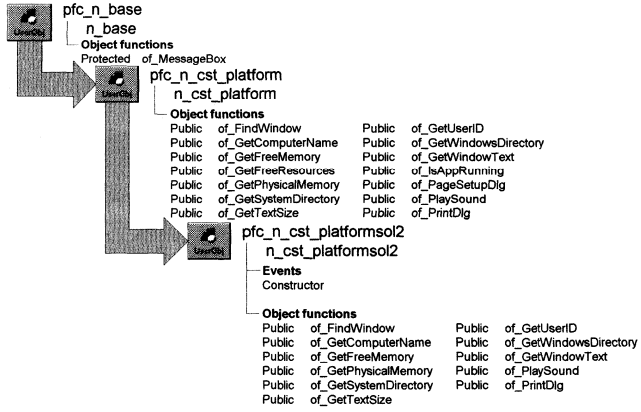
N_cst_platformaix and n_cst_platformhpux provide the identical functionality on IBM AIX and HP/UX.

The functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar.

This object calls functions in the following platform-specific modules:

libgdi31.so
libkernel31.so

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use the platform service:

1 Instantiate the appropriate platform services object by calling the f_SetPlatform global function. This function examines your environment and instantiates the appropriate user object.

2 Call platform services functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_platform) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_memorystatus	ul_availpagefile	UnsignedLong
	ul_availphys	UnsignedLong
	ul_availvirtual	UnsignedLong
	ul_length	UnsignedLong
	ul_memoryload	UnsignedLong
	ul_totalpagefile	UnsignedLong
	ul_totalphys	UnsignedLong
	ul_totalvirtual	UnsignedLong
os_size	l_cx	Long
	l_cy	Long

See also

f_SetPlatform
n_cst_filesrv
n_cst_platform
n_cst_platformmac
n_cst_platformwin16
n_cst_platformwin32

Events

N_cst_platformsol2 includes a precoded event script:

Constructor

Constructor

Description — Initializes the is_separator instance variable.

Usage — This event executes when the object is created.

Functions

N_cst_platformsol2 includes precoded object functions:

of_FindWindow	of_GetUserID
of_GetComputerName	of_GetWindowsDirectory
of_GetFreeMemory	of_GetWindowText
of_GetPhysicalMemory	of_PlaySound
of_GetSystemDirectory	of_PrintDlg
of_GetTextSize	

of_FindWindow

Description — Retrieves the handle of the specified window, as identified by window title.

Access — Public

Syntax — *instancename*.**of_FindWindow** (*windowname*)

Argument	Description
instancename	Instance name of n_cst_platform
windowname	String containing the window title. For MDI applications, this is the title of the frame window

Return value — UnsignedInteger. Returns the handle of *windowname*.

Usage — This function calls the FindWindowA external function.

Examples — This example calls the of_FindWindow function:

```
UnsignedInteger   lui_handle

lui_handle = inv_platform.of_FindWindow &
   (gnv_app.of_GetFrame( ).title)
MessageBox("Platform Service", &
   "Window handle is: " + String(lui_handle) )
```

of_GetComputerName

Description

Retrieves the name of the current workstation.

Access

Public

Syntax

instancename.**of_GetComputerName** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Returns the name of the current workstation.

Usage

This function calls the GetComputerNameA external function.

Examples

This example calls the of_GetComputerName function:

```
String   ls_computer

ls_computer = inv_platform.of_GetComputerName( )
MessageBox("Platform", "Current workstations is:" &
   + ls_computer )
```

of_GetFreeMemory

Description

Retrieves the number of bytes of available memory (including the maximum allowable size of the swap file).

Access

Public

Syntax

instancename.**of_GetFreeMemory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

Long. Returns the number of bytes of available memory.

Usage

This function calls the GlobalMemoryStatus external function.

Examples

This example calls the of_GetFreeMemory function:

```
Long   ll_free

ll_free = inv_platform.of_GetFreeMemory( )
MessageBox("Platform", String(ll_free) &
   + " bytes of free memory" )
```

of_GetPhysicalMemory

Description: Retrieves the total number of bytes of physical memory (RAM).

Access: Public

Syntax: *instancename*.**of_GetPhysicalMemory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value: Long. Returns the number of bytes of memory.

Usage: This function calls the GlobalMemoryStatus external function.

Examples: This example calls the of_GetPhysicalMemory function:

```
Long  ll_total

ll_total = inv_platform.of_GetPhysicalMemory( )
MessageBox("Platform", String(ll_total) &
   + " bytes of RAM (total)" )
```

of_GetSystemDirectory

Description: Retrieves the path of the system directory.

Access: Public

Syntax: *instancename*.**of_GetSystemDirectory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value: String. Returns the path of the system directory if the function succeeds and an empty string if it cannot determine the system directory.

Usage: This function calls the GetSystemDirectoryA external function.

Examples: This example calls the of_GetSystemDirectory function:

```
String  ls_system

ls_system = inv_platform.of_GetSystemDirectory()
IF ls_system <> "" THEN
   MessageBox("Platform", "System Directory is: " &
```

```
        + String(ls_system) )
ELSE
...
```

of_GetTextSize

Description

Calculates height and width in PBUs, based on the passed input arguments.

Access

Public

Syntax

instancename.**of_GetTextSize** (*window, text, font, fontsize, bold, italic, underline, height, width*)

Argument	Description
instancename	Instance name of n_cst_platform
window	Window variable used by the function to create temporary text
text	String specifying the text for which the function calculates height and width
font	String specifying the font
fontsize	Integer specifying the font size
bold	Boolean specifying whether *text* is bold
italic	Boolean specifying whether *text* is italic
underline	Boolean specifying whether *text* is underlined
height	Integer into which the function places the text height, in PBUs (passed by reference)
width	Integer into which the function places the text width, in PBUs (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when sizing text objects and controls manually in a multiplatform environment.

This function calls the GetDC, SelectObject, GetTextExtentPoint32A, and ReleaseDC external functions.

Examples

This example is from the n_cst_dwsrv_report of_AddText function:

```
IF inv_Platform.of_GetTextSize(lw_parent, &
   ls_text + " ", as_fontface, ai_fontsize, &
```

```
    ab_bold, ab_italic, ab_underline, ai_height, &
       ai_width) = -1 THEN Return -1
```

of_GetUserID

Description: Retrieves the current network logon name.

Access: Public

Syntax: *instancename*.**of_GetUserID ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value: String. Returns the current network logon ID if the function succeeds and an empty string if no log on ID can be found.

Usage: This function calls the GetUserNameA external function.

Examples: This example calls the of_GetUserID function:

```
String   ls_userid

ls_userid = inv_platform.of_GetUserID( )
IF ls_userid <> "" THEN
   MessageBox("Platform", "User ID is: " &
      + String(ls_userid) )
ELSE
...
```

of_GetWindowsDirectory

Description: Retrieves the Windows directory.

Access: Public

Syntax: *instancename*.**of_GetWindowsDirectory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value: String. Returns the path of the Windows directory if the function succeeds and an empty string if it cannot determine the Windows directory.

Usage

This function calls the GetWindowsDirectoryA external function.

Examples

This example calls the of_GetWindowsDirectory function:

```
String   ls_windows

ls_windows = inv_platform.of_GetWindowsDirectory()
IF ls_windows <> "" THEN
   MessageBox("Platform", "Windows Directory is:" &
      + String(ls_windows) )
ELSE
...
```

of_GetWindowText

Description

Retrieves the text in the window title.

Access

Public

Syntax

instancename.**of_GetWindowText** (*window*)

Argument	Description
instancename	Instance name of n_cst_platform
window	UnsignedInteger containing the window handle

Return value

String. Returns the title text of the specified window if the function succeeds and an empty string if an error occurs.

Usage

Call this function to determine title text when you know the window handle.

This function calls the GetWindowTextA external function.

Examples

This example calls the of_GetWindowText function:

```
String   ls_title

ls_title = inv_platform.of_GetWindowText &
   (handle(this) )
MessageBox("Platform Service", &
   "Window title is: " + ls_title)
```

of_PlaySound

Description

Plays the specified sound file.

Access

Public

Syntax

instancename.**of_PlaySound** (*filename*)

Argument	Description
instancename	Instance name of n_cst_platform
filename	String specifying the complete pathname of the sound file to play

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the sndPlaySoundA external function.

Examples

This example calls the of_PlaySound function:

```
String   ls_sound

ls_sound=sle_soundfile.text
IF inv_platform.of_PlaySound(ls_sound) = -1 THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      ("No sound capabilities")
END IF
```

of_PrintDlg

Description

Displays an operating-system specific Print dialog box.

Access

Public

Syntax

instancename.**of_PrintDlg** (*printinfo*, *window*)

Argument	Description
instancename	Instance name of n_cst_platform
printinfo	S_printdlgattrib instance containing print information (passed by reference)
window	Window variable containing the window opening the Print dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs or if the user canceled out of the dialog box.

Examples

This example is from the u_dw pfc_PrintDlg event:

```
...
astr_printdlg.b_allpages = TRUE

ls_copies = this.object.datawindow.print.copies
IF NOT IsNumber (ls_copies) THEN
   ls_copies = "1"
END IF
li_copies = Integer(ls_copies)
astr_printdlg.l_copies = li_copies
ls_collate = this.object.datawindow.print.collate
lb_collate = lnv_conversion.of_Boolean(ls_collate)
astr_printdlg.b_collate = lb_collate
astr_printdlg.l_frompage = 1
astr_printdlg.l_minpage = 1
...
this.Event pfc_PrePrintDlg(astr_printdlg)

f_SetPlatform(lnv_platform, TRUE)
this.of_GetParentWindow(lw_parent)
ll_rc = &
   lnv_platform.of_PrintDlg &
astr_printdlg, lw_parent)
f_SetPlatform(lnv_platform, FALSE)
...
```

n_cst_platformwin16

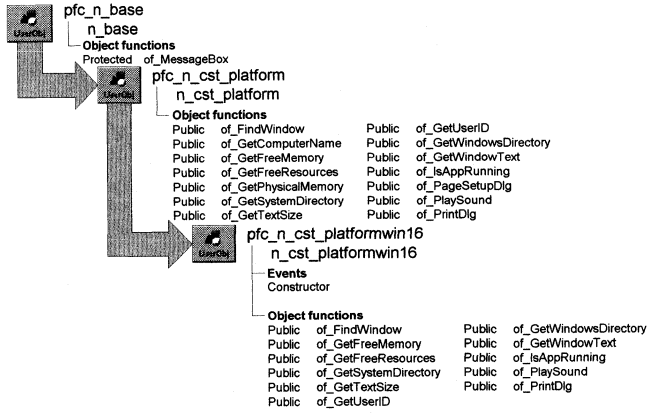
Description

Platform-specific functions for applications running under Windows 3.1. Platform-specific functions require a call to a platform-specific external function. When you call the f_SetPlatform function, it detects the operating environment and instantiates the correct platform-specific object.

The functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar. This object calls functions in the following platform-specific Windows 3.1 modules:

GDI.EXE
KERNEL.EXE
MMSYSTEM.DLL
PFCCOMM.DLL
USER.EXE

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use the platform service:

1 Instantiate the appropriate platform services object by calling the f_SetPlatform global function. This function examines your environment and instantiates the appropriate user object.

2 Call platform services functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_platform) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

See also

f_SetPlatform
n_cst_filesrv
n_cst_platform
n_cst_platformmac
n_cst_platformsol2
n_cst_platformwin32

Events

N_cst_platformwin16 includes a precoded event script:

Constructor

Constructor

Description

Initializes the is_separator instance variable.

Usage

This event executes when the object is created.

Functions

N_cst_platformwin16 includes precoded object functions:

of_FindWindow	of_GetWindowsDirectory
of_GetFreeMemory	of_GetWindowText
of_GetFreeResources	of_IsAppRunning
of_GetSystemDirectory	of_PlaySound
of_GetTextSize	of_PrintDlg
of_GetUserID	

of_FindWindow

Description: Retrieves the handle of the specified window, as identified by window title.

Access: Public

Syntax: *instancename*.**of_FindWindow** (*windowname*)

Argument	Description
instancename	Instance name of n_cst_platform
windowname	String containing the window title. For MDI applications, this is the title of the frame window

Return value: UnsignedInteger. Returns the handle of *windowname*.

Usage: This function calls the FindWindow external function.

Examples: This example calls the of_FindWindow function:

```
UnsignedInteger  lui_handle

lui_handle = inv_platform.of_FindWindow &
   (gnv_app.of_GetFrame( ).title)
MessageBox("Platform Service", &
   "Window handle is: " + String(lui_handle) )
```

of_GetFreeMemory

Description: Retrieves the number of bytes of available memory (including the maximum allowable size of the swap file).

Access: Public

Syntax: *instancename*.**of_GetFreeMemory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value: Long. Returns the number of bytes of available memory.

Usage: This function calls the GetFreeSpace external function.

Examples: This example calls the of_GetFreeMemory function:

```
Long  ll_free
```

```
ll_free = inv_platform.of_GetFreeMemory()
MessageBox("Platform", String(ll_free) &
   + " bytes of free memory" )
```

of_GetFreeResources

Description: Retrieves the percentage of free space for system resources.

Access: Public

Syntax: *instancename*.**of_GetFreeResources** (*type*)

Argument	Description
instancename	Instance name of n_cst_platform
type	Integer specifying the resource type to return. Values are: ♦ 0 System resources ♦ 1 GDI resources ♦ 2 User resources

Return value: Integer. Returns the percentage of free space for system resources.

Usage: This function calls the GetFreeSystemResources external function.

Examples: This example calls the of_GetFreeResources function:

```
Integer   li_free, li_arg

li_arg = 1
li_free = &
   inv_platform.of_GetFreeResources(li_arg)
MessageBox("Platform", String(li_free) &
   + " percent free system resources" )
```

of_GetSystemDirectory

Description: Retrieves the path of the system directory (for example, Windows\System).

Access: Public

Syntax: *instancename*.**of_GetSystemDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Returns the path of the system directory if the function succeeds and an empty string if it cannot determine the system directory.

Usage

This function calls the GetSystemDirectory external function.

Examples

This example calls the of_GetSystemDirectory function:

```
String   ls_system

ls_system = inv_platform.of_GetSystemDirectory()
IF ls_system <> "" THEN
   MessageBox("Platform", "System Directory is: " &
      + String(ls_system) )
ELSE
...
```

of_GetTextSize

Description

Calculates height and width in PBUs, based on the passed input arguments.

Access

Public

Syntax

instancename.**of_GetTextSize** (*window, text, font, fontsize, bold, italic, underline, height, width*)

Argument	Description
instancename	Instance name of n_cst_platform
window	Window variable used by the function to create temporary text
text	String specifying the text for which the function calculates height and width
font	String specifying the font
fontsize	Integer specifying the font size
bold	Boolean specifying whether *text* is bold
italic	Boolean specifying whether *text* is italic
underline	Boolean specifying whether *text* is underlined

Argument	Description
height	Integer into which the function places the text height, in PBUs (passed by reference)
width	Integer into which the function places the text width, in PBUs (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when sizing text objects and controls manually in a multiplatform environment.

This function calls the GetDC, SelectObject, ReleaseDC, and GetTextExtent external functions.

Examples

This example is from the n_cst_dwsrv_report of_AddText function:

```
IF inv_Platform.of_GetTextSize(lw_parent, &
   ls_text + " ", as_fontface, ai_fontsize, &
      ab_bold, ab_italic, ab_underline, ai_height, &
         ai_width) = -1 Then Return -1
```

of_GetUserID

Description

Retrieves the current network logon name.

Access

Public

Syntax

instancename.**of_GetUserID** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Returns the current network logon ID if the function succeeds and an empty string if no logon ID can be found.

Usage

This function calls the WNetGetUser external function.

Examples

This example calls the of_GetUserID function:

```
String   ls_userid

ls_userid = inv_platform.of_GetUserID( )
IF ls_userid <> "" THEN
   MessageBox("Platform", "User ID is: " &
```

```
        + String(ls_userid) )
ELSE
...
```

of_GetWindowsDirectory

Description — Retrieves the Windows directory (for example, C:\Windows).

Access — Public

Syntax — *instancename*.**of_GetWindowsDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value — String. Returns the path of the Windows directory if the function succeeds and an empty string if it cannot determine the Windows directory.

Usage — This function calls the GetWindowsDirectory external function.

Examples — This example calls the of_GetWindowsDirectory function:

```
String   ls_windows

ls_windows = inv_platform.of_GetWindowsDirectory( )
IF ls_windows <> "" THEN
   MessageBox("Platform", "Windows Directory is:" &
      + String(ls_windows) )
ELSE
...
```

of_GetWindowText

Description — Retrieves the text in the window title.

Access — Public

Syntax — *instancename*.**of_GetWindowText** (*window*)

Argument	Description
instancename	Instance name of n_cst_platform
window	UnsignedInteger containing the window handle

Return value

String. Returns the title text of the specified window.

Usage

This function calls the GetWindowText external function.

Examples

This example calls the of_GetWindowText function:

```
String   ls_title

ls_title = inv_platform.of_GetWindowText &
   (Handle(this) )
MessageBox("Platform Service", &
   "Window title is: " + ls_title)
```

of_IsAppRunning

Description

Reports whether an application is currently running.

Access

Public

Syntax

instancename.**of_IsAppRunning** (*appname*)

Argument	Description
instancename	Instance name of n_cst_platform
appname	String specifying the application to check for

Return value

Boolean. Returns TRUE if *appname* is running and FALSE if it is not.

Usage

This function calls the GetModuleHandle and GetModuleUsage external functions.

Examples

This example calls the of_IsAppRunning function:

```
IF NOT inv_platform.of_IsAppRunning &
   ("Notepad") THEN
      Run("Notepad")
ELSE
   gnv_app.of_GetFrame().SetMicroHelp &
      ("Notepad is already running")
END IF
```

of_PlaySound

Description

Plays the specified sound file.

Access

Public

Syntax

instancename.**of_PlaySound** (*filename*)

Argument	Description
instancename	Instance name of n_cst_platform
filename	String specifying the complete pathname of the sound file to play

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the WaveOutGetNumDevs and sndPlaySound external functions.

Examples

This example calls the of_PlaySound function:

```
String   ls_sound

ls_sound=sle_soundfile.text
IF inv_platform.of_PlaySound(ls_sound) = -1 THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      ("No sound capabilities")
END IF
```

of_PrintDlg

Description

Displays an operating-system specific Print dialog box.

Access

Public

Syntax

instancename.**of_PrintDlg** (*printinfo, window*)

Argument	Description
instancename	Instance name of n_cst_platform
printinfo	S_printdlgattrib instance containing print information (passed by reference)
window	Window variable containing the window opening the Print dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs or if the user canceled out of the dialog box.

Examples

This example is from the u_dw pfc_PrintDlg event:

```
...
```

```
astr_printdlg.b_allpages = TRUE

ls_copies = this.object.datawindow.print.copies
IF NOT IsNumber (ls_copies) THEN
   ls_copies = "1"
END IF
li_copies = Integer(ls_copies)
astr_printdlg.l_copies = li_copies
ls_collate = this.object.datawindow.print.collate
lb_collate = lnv_conversion.of_Boolean(ls_collate)
astr_printdlg.b_collate = lb_collate
astr_printdlg.l_frompage = 1
astr_printdlg.l_minpage = 1
...
this.Event pfc_PrePrintDlg(astr_printdlg)

f_SetPlatform(lnv_platform, TRUE)
this.of_GetParentWindow(lw_parent)
ll_rc = &
   lnv_platform.of_PrintDlg &
      (astr_printdlg, lw_parent)
f_SetPlatform(lnv_platform, FALSE)
...
```

n_cst_platformwin32

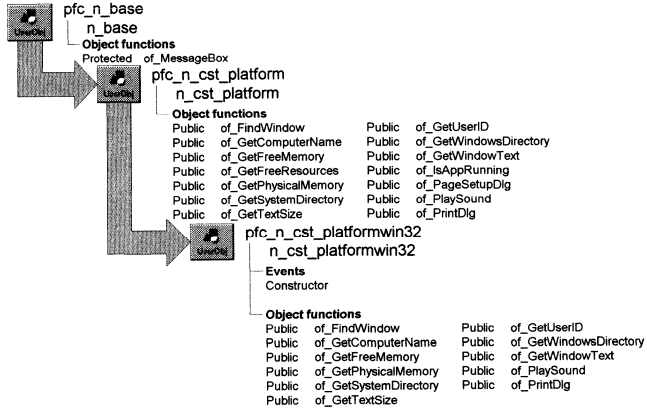
Description

Platform-specific functions for applications running on Windows 95 and Windows NT. Platform-specific functions require a call to a platform-specific external function. When you call the f_SetPlatform function, it detects the operating environment and instantiates the correct platform-specific object.

The functions in this object call platform-specific external functions. To see a complete list of platform-specific external functions used by this object, access the User Object painter and select Declare>Local External Functions from the menu bar. This object calls functions in the following platform-specific 32-bit Windows modules:

ADVAPI32.EXE
GDI32.EXE
KERNEL32.EXE
PFCCOM32.EXE
USER32.EXE
WINMM.DLL

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use the platform service:

1 Instantiate the appropriate platform services object by calling the f_SetPlatform global function. This function examines your environment and instantiates the appropriate user object.

2 Call platform services functions as required. The code for these functions is defined in the descendant objects. If the function isn't available on your platform, PowerBuilder executes the ancestor (n_cst_platform) function, which, depending on the function, returns an empty string, TRUE, -1, 0, or NULL.

Structures

Structure	Field	Data type
os_memorystatus	ul_availpagefile	UnsignedLong
	ul_availphys	UnsignedLong
	ul_availvirtual	UnsignedLong
	ul_length	UnsignedLong
	ul_memoryload	UnsignedLong
	ul_totalpagefile	UnsignedLong
	ul_totalphys	UnsignedLong
	ul_totalvirtual	UnsignedLong
os_size	l_cx	Long
	l_cy	Long

See also

f_SetPlatform
n_cst_filesrv
n_cst_platform
n_cst_platformwin16

Events

N_cst_platformwin32 includes a precoded event script:

Constructor

Constructor

Description

Initializes the is_separator instance variable.

Usage

This event executes when the object is created.

Functions

N_cst_platformwin32 includes precoded object functions:

of_FindWindow	of_GetUserID
of_GetComputerName	of_GetWindowsDirectory
of_GetFreeMemory	of_GetWindowText
of_GetPhysicalMemory	of_PlaySound
of_GetSystemDirectory	of_PrintDlg
of_GetTextSize	

of_FindWindow

Description

Returns the handle of the specified window, as identified by window title.

Access

Public

Syntax

instancename.**of_FindWindow** (*windowname*)

Argument	Description
instancename	Instance name of n_cst_platform
windowname	String containing the window title. For MDI applications, this is the title of the frame window

Return value

UnsignedInteger. Returns the handle of *windowname*.

Usage

This function calls the FindWindowA external function.

Examples

This example calls the of_FindWindow function:

```
UnsignedInteger lui_handle

lui_handle = inv_platform.of_FindWindow &
   (gnv_app.of_GetFrame( ).title)
MessageBox("Platform Service", &
   "Window handle is: " + String(lui_handle) )
```

of_GetComputerName

Description

Returns the name of the current workstation.

Access

Public

Syntax

instancename.**of_GetComputerName** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

String. Returns the name of the current workstation.

Usage

This function calls the GetComputerNameA external function.

Examples

This example calls the of_GetComputerName function:

```
String   ls_computer

ls_computer = inv_platform.of_GetComputerName()
MessageBox("Platform", "Current workstations is:" &
   + ls_computer )
```

of_GetFreeMemory

Description

Returns the number of bytes of available memory (including the maximum allowable size of the swap file).

Access

Public

Syntax

instancename.**of_GetFreeMemory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value

Long. Returns the number of bytes of available memory.

Usage

This function calls the GlobalMemoryStatus external function.

Examples

This example calls the of_GetFreeMemory function:

```
Long   ll_free

ll_free = inv_platform.of_GetFreeMemory()
MessageBox("Platform", String(ll_free) &
   + " bytes of free memory" )
```

of_GetPhysicalMemory

Description: Returns the total number of bytes of physical memory (RAM).

Access: Public

Syntax: *instancename*.**of_GetPhysicalMemory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value: Long. Returns the number of bytes of memory.

Usage: This function calls the GlobalMemoryStatus external function.

Examples: This example calls the of_GetPhysicalMemory function:

```
Long  ll_total

ll_total = inv_platform.of_GetPhysicalMemory( )
MessageBox("Platform", String(ll_total) &
   + " bytes of RAM (total)" )
```

of_GetSystemDirectory

Description: Returns the path of the system directory (for example, Windows\System).

Access: Public

Syntax: *instancename*.**of_GetSystemDirectory ( )**

Argument	Description
instancename	Instance name of n_cst_platform

Return value: String. Returns the path of the system directory if the function succeeds and an empty string if it cannot determine the system directory.

Usage: This function calls the GetSystemDirectoryA external function.

Examples: This example calls the of_GetSystemDirectory function:

```
String   ls_system

ls_system = inv_platform.of_GetSystemDirectory()
IF ls_system <> "" THEN
   MessageBox("Platform", "System Directory is: " &
```

```
        + String(ls_system) )
ELSE
...
```

of_GetTextSize

Description

Calculates height and width in PBUs, based on the passed input arguments.

Access

Public

Syntax

instancename.**of_GetTextSize** (*window, text, font, fontsize, bold, italic, underline, height, width*)

Argument	Description
instancename	Instance name of n_cst_platform
window	Window variable used by the function to create temporary text
text	String specifying the text for which the function calculates height and width
font	String specifying the font
fontsize	Integer specifying the font size
bold	Boolean specifying whether *text* is bold
italic	Boolean specifying whether *text* is italic
underline	Boolean specifying whether *text* is underlined
height	Integer into which the function places the text height, in PBUs (passed by reference)
width	Integer into which the function places the text width, in PBUs (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when sizing text objects and controls manually in a multiplatform environment.

This function calls the GetDC, SelectObject, GetTextExtentPoint32A, and ReleaseDC external functions.

Examples

This example is from the n_cst_dwsrv_report of_AddText function:

```
IF inv_Platform.of_GetTextSize(lw_parent, &
   ls_text + " ", as_fontface, ai_fontsize, &
```

```
        ab_bold, ab_italic, ab_underline, ai_height, &
           ai_width) = -1 Then Return -1
```

of_GetUserID

Description — Returns the current network logon name.

Access — Public

Syntax — *instancename*.**of_GetUserID** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value — String. Returns the current network logon ID if the function succeeds and an empty string if no log on ID can be found.

Examples — This example calls the of_GetUserID function:

```
String   ls_userid

ls_userid = inv_platform.of_GetUserID( )
IF ls_userid <> "" THEN
   MessageBox("Platform", "User ID is: " &
      + String(ls_userid) )
ELSE
...
```

of_GetWindowsDirectory

Description — Returns the Windows directory (for example, C:\Windows).

Access — Public

Syntax — *instancename*.**of_GetWindowsDirectory** ( )

Argument	Description
instancename	Instance name of n_cst_platform

Return value — String. Returns the path of the Windows directory if the function succeeds and an empty string if it cannot determine the Windows directory.

Usage — This function calls the GetWindowsDirectoryA external function.

Examples

This example calls the of_GetWindowsDirectory function:

```
String   ls_windows

ls_windows = inv_platform.of_GetWindowsDirectory()
IF ls_windows <> "" THEN
   MessageBox("Platform", "Windows Directory is:" &
      + String(ls_windows) )
ELSE
...
```

of_GetWindowText

Description

Returns the text in the window title.

Access

Public

Syntax

instancename.**of_GetWindowText** (*window*)

Argument	Description
instancename	Instance name of n_cst_platform
window	UnsignedInteger containing the window handle

Return value

String. Returns the title text of the specified window if the function succeeds and an empty string if an error occurs.

Usage

This function calls the GetWindowTextA external function.

Examples

This example calls the of_GetWindowText function:

```
String   ls_title

ls_title = inv_platform.of_GetWindowText &
   (handle(this) )
MessageBox("Platform Service", &
   "Window title is: " + ls_title)
```

of_PlaySound

Description

Plays the specified sound file.

Access

Public

Syntax

instancename.**of_PlaySound** (*filename*)

Argument	Description
instancename	Instance name of n_cst_platform
filename	String specifying the complete pathname of the sound file to play

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function calls the WaveOutGetNumDevs and sndPlaySoundA external functions.

Examples

This example calls the of_PlaySound function:

```
String   ls_sound

ls_sound=sle_soundfile.text
IF inv_platform.of_PlaySound(ls_sound) = -1 THEN
   gnv_app.of_GetFrame().SetMicroHelp &
      ("No sound capabilities")
END IF
```

of_PrintDlg

Description

Displays an operating-system specific Print dialog box.

Access

Public

Syntax

instancename.**of_PrintDlg** (*printinfo*, *window*)

Argument	Description
instancename	Instance name of n_cst_platform
printinfo	S_printdlgattrib instance containing print information (passed by reference)
window	Window variable containing the window opening the Print dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs or if the user canceled out of the dialog box.

Examples

This example is from the u_dw pfc_PrintDlg event:

```
...
astr_printdlg.b_allpages = TRUE
```

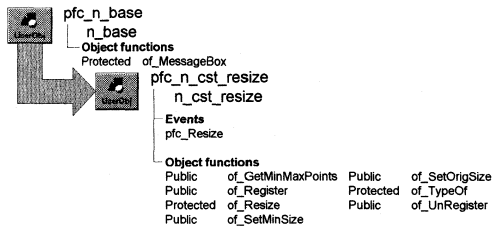
```
ls_copies = this.object.datawindow.print.copies
IF NOT IsNumber (ls_copies) THEN
   ls_copies = "1"
END IF
li_copies = Integer(ls_copies)
astr_printdlg.l_copies = li_copies
ls_collate = this.object.datawindow.print.collate
lb_collate = lnv_conversion.of_Boolean(ls_collate)
astr_printdlg.b_collate = lb_collate
astr_printdlg.l_frompage = 1
astr_printdlg.l_minpage = 1
...
this.Event pfc_PrePrintDlg(astr_printdlg)

f_SetPlatform(lnv_platform, TRUE)
this.of_GetParentWindow(lw_parent)
ll_rc = &
   lnv_platform.of_PrintDlg &
      astr_printdlg, lw_parent)
f_SetPlatform(lnv_platform, FALSE)
...
```

n_cst_resize

Description

Resize service object.

Ancestry



Library

PFCWNSRV.PBL
PFEWNSRV.PBL

Usage

To use this service:

1 Enable the service using the w_master, u_tab, or u_tabpg of_SetResize function. This function automatically sets the original size instance variables:

```
this.of_SetResize(TRUE)
```

2 (Optional) Specify the window, tab, or tab page's original size by calling the of_SetOrigSize function. You call this function if an MDI application opens MDI sheets with an enumeration other than Original!:

```
this.inv_resize.of_SetOrigSize &
(this.width, this.height)
```

3 (Optional) Call the of_SetMinSize function to specify a minimum size below which the resize service no longer resizes registered controls:

```
this.inv_resize.of_SetMinSize &
   (this.width-50, this.height-100)
```

4 Call the of_Register function once for each control that you want to be resized automatically whenever the object resizes.

```
this.inv_resize.of_Register &
    (lv_emp, this.inv_resize.SCALERIGHTBOTTOM)
this.inv_resize.of_Register &
    (cb_ok, this.inv_resize.FIXEDRIGHT)
this.inv_resize.of_Register &
```

```
(cb_cancel, this.inv_resize.FIXEDRIGHT)
```

The Resize event calls the n_cst_resize of_Resize function to perform the resizing.

5 (Optional) Call the of_UnRegister function to remove controls from the resize list.

Open MDI sheets using the Original! enumeration

If you open MDI sheets with an enumeration other than Original!, call the of_SetOrigSize function before registering controls with the resize service. The call to of_SetOrigSize passes what the workspace width and height would have been had the sheet opened in the original size. This applies to controls in a window, tab, or tab page.

Structures

Structure	Field	Data type
os_resize	b_movex	Boolean
	b_movey	Boolean
	b_scale	Boolean
	b_scaleheight	Boolean
	b_scalewidth	Boolean
	i_heightmin	Integer
	i_movex	Integer
	i_movey	Integer
	i_scaleheight	Integer
	i_scalewidth	Integer
	i_widthmin	Integer
	r_height	Real
	r_width	Real
	r_x	Real
	r_y	Real
	s_classname	String
	s_typeof	String

Structure	Field	Data type
	wo_control	GraphicObject

See also

n_cst_dwsrv_resize
u_tab
u_tabpg
w_master

Instance variables

Instance variable	Description	Data type	Access	Usage
DRAGOBJECT	Constant set to DragObject!	String	Protected	Internal
FIXEDBOTTOM	Constant set to FixedToBottom	String	Public	Use with of_Register
FIXEDBOTTOM_SCALERIGHT	Constant set to FixedToBottom& ScaleToRight	String	Public	Use with of_Register
FIXEDRIGHT	Constant set to FixedToRight	String	Public	Use with of_Register
FIXEDRIGHT_SCALEBOTTOM	Constant set to FixedToRight& ScaleToBottom	String	Public	Use with of_Register
FIXEDRIGHTBOTTOM	Constant set to FixedToRight&Bottom	String	Public	Use with of_Register
ii_rounding	Rounding factor	Integer	Protected	Internal (default is 5)
il_parentminimumheight	Minimum height for the parent window	Long	Protected	Set with the of_SetMinSize
il_parentminimumwidth	Minimum width for the parent window	Long	Protected	Set with the of_SetMinSize
il_parentprevheight	Original height of the parent window	Long	Protected	Set with of_SetOrigSize
il_parentprevwidth	Original width of the parent window	Long	Protected	Set with of_SetOrigSize

Instance variable	Description	Data type	Access	Usage
istr_registered	Structure array to track objects eligible for resizing	os_resize	Protected	Internal
LINE	Constant set to Line!	String	Protected	Internal
MDICLIENT	Constant set to MDIClient!	String	Protected	Internal
OVAL	Constant set to Oval!	String	Protected	Internal
RECTANGLE	Constant set to Rectangle!	String	Protected	Internal
ROUNDRECTANGLE	Constant set to RoundRectangle!	String	Protected	Internal

Events

N_cst_resize includes a precoded event:

pfc_Resize

pfc_Resize

Description

Calls the of_Resize function, resizing the object, as specified in the of_Register function calls.

Syntax

instancename.EVENT **pfc_Resize** (*sizetype*, *newwidth*, *newheight*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
sizetype	UnsignedLong specifying the resize type: ♦ 0 — (Restored) The window or DataWindow has been resized, but it was not minimized or maximized. The user may have dragged the borders or a script may have called the Resize PowerScript function ♦ 1 — (Minimized) The window or DataWindow has been minimized ♦ 2 — (Maximized) The window or DataWindow has been maximized

Argument	Description
newwidth	Integer specifying the new width
newheight	Integer specifying the new height

Usage

The u_dw, u_tab, and u_tabpg Resize events call this event.

Functions

N_cst_resize includes precoded object functions:

of_GetMinMaxPoints	of_SetOrigSize
of_Register	of_TypeOf
of_Resize	of_UnRegister
of_SetMinSize	

of_GetMinMaxPoints

Description

Retrieves the four extreme points of all controls within a window.

Access

Public

Syntax

instancename.**of_GetMinMaxPoints** (*controls*, *minx*, *miny*, *maxx*, *maxy*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
controls	WindowObject array containing the controls for which the minimum and maximum points are returned
minx	Integer into which the function places the minimum *x* coordinate (passed by reference)
miny	Integer into which the function places the minimum *y* coordinate (passed by reference)
maxx	Integer into which the function places the maximum *x* coordinate (passed by reference)
maxy	Integer into which the function places the maximum *y* coordinate (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetMinMaxPoints function from within a window control:

```
WindowObject   lwo_controls[ ]
Integer  li_minx, li_miny, li_maxx, li_maxy
Integer  li_return

li_return = parent.inv_resize.of_GetMinMaxPoints &
   (parent.control[ ], li_minx, li_miny, &
     li_maxx, li_maxy)
IF li_return = -1 THEN
   MessageBox("Window Resize", &
      "Error in of_GetMinMaxPoints")
ELSE
   MessageBox("Window Resize", &
      "Minimum x: " + String(li_minx) &
      + "~r~nMinimum y: " + String(li_miny) &
      + "~r~nMaximum x: " + String(li_maxx) &
      + "~r~nMaximum y: " + String(li_maxy))
END IF
```

of_Register

Registers a control for resizing. The resize service moves and optionally resizes registered controls when the user resizes the window. There are two syntaxes:

To register by specifying	Use
Resize and move percentages	Syntax 1
Resize method	Syntax 2

One additional syntax

N_cst_resize supports one additional of_Register syntax, which is defined as protected and used internally by the other syntaxes.

For more information on this syntax, use the Object Browser.

Syntax 1

Register by specifying resize and move percentages

Description

Registers a control for resizing given resize and move percentages.

Access

Public

Syntax

instancename.**of_Register** (*control*, *movex*, *movey*, *scalewidth*, *scaleheight*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
control	WindowObject variable specifying the object to be resized
movex	Integer specifying the percentage to move the object along the x axis
movey	Integer specifying the percentage to move the object along the y axis
scalewidth	Integer specifying the percentage to resize the object along the x axis
scaleheight	Integer specifying the percentage to resize the object along the y axis

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Because you can explicitly specify resize and move percentages, this syntax provides more flexibility than Syntax 2.

Examples

This example calls the of_Register function:

```
this.inv_resize.of_Register &
 (dw_1, inv_resize.0, 0, 50, 50)
this.inv_resize.of_Register &
  (dw_2, 100, 0, 50, 50)
```

Syntax 2

Register by specifying a resize method

Description

Registers a control for resizing given a resize method.

Access

Public

Syntax

instancename.**of_Register** (*control*, *method*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
control	WindowObject variable containing the control to be registered

Argument	Description
method	String or n_cst_resize constant specifying how the resize service handles *control* when the window Resize event occurs: ♦ "FixedToRight" or FIXEDRIGHT ♦ "FixedToBottom" or FIXEDBOTTOM ♦ "FixedToRight&Bottom" or FIXEDRIGHTBOTTOM ♦ "Scale" or SCALE ♦ "ScaleToRight" or SCALERIGHT ♦ "ScaleToBottom" or SCALEBOTTOM ♦ "ScaleToRight&Bottom" or SCALERIGHTBOTTOM ♦ "FixedToRight&ScaleToBottom" or FIXEDRIGHT_SCALEBOTTOM ♦ "FixedToBottom&ScaleToRight" or FIXEDBOTTOM_SCALERIGHT

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Register function:

```
this.of_SetResize(TRUE)
this.inv_resize.of_SetOrigSize &
   (this.width, this.height)
this.inv_resize.of_Register &
    (lv_emp, this.inv_resize.SCALERIGHTBOTTOM)
this.inv_resize.of_Register &
    (cb_ok, this.inv_resize.FIXEDRIGHT)
this.inv_resize.of_Register &
    (cb_cancel, this.inv_resize.FIXEDRIGHT)
```

of_Resize

Description

Moves or resizes registered controls, as specified in the of_Register function.

Access

Protected

Syntax

instancename.**of_Resize** (*newwidth*, *newheight*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
newwidth	Integer specifying the new window width

Argument	Description
newheight	Integer specifying the new window height

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The w_master Resize event calls this function when the resize service has been enabled.

Examples

This example is from the w_master Resize event:

```
IF IsValid (inv_resize) THEN
   inv_resize.of_Resize &
      (this.WorkSpaceWidth(), &
         this.WorkSpaceHeight())
END IF
```

of_SetMinSize

Description

Establishes a minimum window size below which the resize service no longer modifies the position of registered controls.

Access

Public

Syntax

instancename.**of_SetMinSize** (*minimumwidth, minimumheight*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
minimumwidth	Integer specifying minimum width
minimumheight	Integer specifying minimum height

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMinSize function:

```
this.inv_resize.of_SetMinSize &
   (this.width, this.height)
this.inv_resize.of_Register &
    (lv_emp, this.inv_resize.SCALERIGHTBOTTOM)
this.inv_resize.of_Register &
    (cb_ok, this.inv_resize.FIXEDRIGHT)
this.inv_resize.of_Register &
    (cb_cancel, this.inv_resize.FIXEDRIGHT)
```

of_SetOrigSize

Description

Saves the current window size.

Access

Public

Syntax

instancename.**of_SetOrigSize** (*width*, *height*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
width	Integer specifying the window width
height	Integer specifying the window height

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You call this function before calling the of_Register function.

Examples

This example calls the of_SetOrigSize function:

```
Integer  li_return

this.of_SetResize(TRUE)
li_return = this.inv_resize.of_SetOrigSize &
   (this.width, this.height)
...
```

of_TypeOf

Description

Determines the object type for a window control.

Access

Protected

Syntax

instancename.**of_TypeOf** (*control*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
control	WindowObject variable for which the function determines the object type

Return value

String. Returns the object type if the function succeeds and ! if an error occurs.

Usage

Internal.

Examples

This example is from the of_Register function:

```
Choose Case of_TypeOf(awo_control[li_cnt])
   Case DRAGOBJECT
      ...
```

of_UnRegister

Description

Removes a control from the list of controls eligible for resizing.

Access

Public

Syntax

instancename.**of_UnRegister** (*control*)

Argument	Description
instancename	Instance name of n_cst_resize (the default is inv_resize)
control	WindowObject variable containing the control to be unregistered

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_UnRegister function:

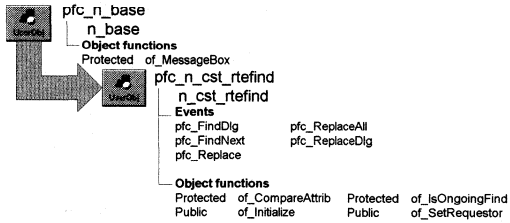
```
Integer  li_return

li_return = parent.inv_resize.of_UnRegister &
   (lv_emp)
IF li_return = -1 THEN
   MessageBox("Resize", "UnRegister failed")
END IF
```

n_cst_rtefind

Description

Find and replace service. Controls find and replace operations in a u_rte-based RichTextEdit control.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

u_rte

Usage

U_rte uses this object to control find and replace processing.

To use this service:

1 Enable the service using the u_rte of_SetFind function:

```
rte_doc.of_SetFind(TRUE)
```

When users select Edit>Find or Edit>Replace from the menu bar of a menu that descends from the m_master menu, the service displays the w_find or w_replace dialog box. You can also display these dialog boxes programmatically, as explained in the next step.

2 Display the w_find dialog box by calling the u_rte pfc_FindDlg event; display the w_replace dialog box by calling the u_rte pfc_ReplaceDlg event. This example displays the w_find dialog box:

```
rte_doc.Event pfc_FindDlg()
```

See also

u_rte

Instance variables

Instance variable	Description	Data type	Access	Usage
ib_ongoingfind	Indicates whether a find operation is a new find or a find next operation	Boolean	Protected	Internal
idw_requestor	U_dw-based DataWindow control that is associated with this instance of n_cst_rtefind	u_dw	Protected	The DataWindow must use the RichTextEdit presentation style. You must add supporting code to implement find and replace for RichTextEdit DataWindows
inv_findattrib	Structure object containing find and replace information	n_cst_findattrib	Protected	Internal
irte_requestor	U_rte-based RichTextEdit control that is associated with this instance of n_cst_rtefind	u_rte	Protected	Internal

Events

N_cst_rtefind includes precoded event scripts:

pfc_FindDlg	pfc_ReplaceAll
pfc_FindNext	pfc_ReplaceDlg
pfc_Replace	

pfc_FindDlg

Description

Displays the w_find dialog box.

Usage

This event is called when the RichTextEdit has focus and the user selects Edit>Find from a menu that descends from the PFC m_master menu.

pfc_FindNext

Description

Begins a search using the current settings.

Syntax

instancename.Event **pfc_FindNext** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_rtefind
findattrib	N_cst_findattrib instance containing find information. From within the pfc_FindNext event, access this value through the *anv_findattrib* argument

Return value

Integer. Returns the number of characters found. This event returns 0 if no matching text is found and -1 if an error occurs.

Usage

This event is called by the w_find and w_replace dialog boxes to continue a search for a previously specified string.

pfc_Replace

Description

Replaces text as specified in the w_replace dialog box.

Syntax

instancename.Event **pfc_Replace** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_rtefind
findattrib	N_cst_findattrib instance containing find information. From within the pfc_Replace event, access this value through the *anv_findattrib* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The w_replace dialog box calls this event when the user clicks Replace.

pfc_ReplaceAll

Description

Calls the pfc_Replace event to replace all occurrences of the search string with the target string.

Syntax

instancename.Event **pfc_ReplaceAllt** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_rtefind
findattrib	n_cst_findattrib instance containing find information. From within the pfc_ReplaceAll event, access this value through the *anv_findattrib* argument

Return value — Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage — The w_replace dialog box calls this event when the user clicks Replace All.

pfc_ReplaceDlg

Description — Displays the w_replace dialog box.

Usage — This event is called when the RichTextEdit has focus and the user selects Edit>Replace from a menu that descends from the PFC m_master menu.

Functions

N_cst_rtefind includes precoded object functions:

of_CompareAttrib	of_IsOngoingFind
of_Initialize	of_SetRequestor

of_CompareAttrib

Description — Reports whether the user has made changes to specifications on the w_replace dialog box.

Access — Protected

Syntax — *instancename*.**of_CompareAttrib** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_rtefind
findattrib	N_cst_findattrib instance containing find information

Return value — Boolean. Returns TRUE if there are no changes and FALSE if changes were made.

Usage — The of_IsOngoingFind function calls this function.

Examples — This example is from the of_IsOnGoingFind function:

```
IF IsValid(anv_findattrib) THEN
   IF NOT of_CompareAttrib(anv_findattrib) THEN
   Return FALSE
```

```
    END IF
END IF
Return TRUE
```

of_Initialize

Description

Initializes the inv_findattrib instance with the passed values.

Access

Public

Syntax

instancename.**of_Initialize** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_rtefind
findattrib	N_cst_findattrib instance containing find information

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function when the inv_findattrib default values are not appropriate.

Examples

This example calls the of_Initialize function:

```
// Display the Find/Replace windows with
// The Direction Control only.
n_cst_findattrib  lnv_findattrib

lnv_findattrib.is_find = ''
lnv_findattrib.is_replacewith = ''
lnv_findattrib.ib_wholewordvisible =False
lnv_findattrib.ib_wholewordenabled =False
lnv_findattrib.ib_wholeword = False
lnv_findattrib.ib_matchcasevisible = False
lnv_findattrib.ib_matchcaseenabled = False
lnv_findattrib.ib_matchcase = False
lnv_findattrib.ib_directionvisible = True
lnv_findattrib.ib_directionenabled = True
lnv_findattrib.is_direction = 'Down'
lnv_findattrib.ib_lookvisible = False
lnv_findattrib.ib_lookenabled = False

rte_emp.inv_find.of_Initialize(lnv_findattrib)
```

of_IsOngoingFind

Description

Reports whether the current find operation is a new Find or Find Next operation.

Access

Public

Syntax

instancename.**of_IsOngoingFind** (*findattrib*)

Argument	Description
instancename	Instance name of n_cst_rtefind
findattrib	N_cst_findattrib instance containing find information

Return value

Boolean. Returns TRUE if the current find is the same as before and FALSE if it is not.

Usage

N_cst_rtefind events call this function to determine is the current find operation is a new Find or Find Next operation.

Examples

This example is from the pfc_FindNext event:

```
...
IF ib_ongoingfind THEN
   ib_ongoingfind = of_IsOngoingFind(anv_findattrib)
END IF
...
```

of_SetRequestor

Description

Associates a RichTextEdit or DataWindow control with this instance of n_cst_rtefind.

Access

Public

Syntax

instancename.**of_SetRequestor** (*requestor*)

Argument	Description
instancename	Instance name of n_cst_rtefind
requestor	U_rte-based RichTextEdit control or u_dw-based DataWindow window control to associate with this instance of n_cst_rtefind

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The u_rte of_SetFind function calls this function.

If you are writing an extension to use this service with a DataWindow using the RichTextEdit presentation style, call this function to associate the DataWindow with this service.

Examples

This example is from the u_rte of_SetFind function:

```
...
IF ab_Switch THEN
   IF Not IsValid(inv_find) THEN
      inv_find = Create n_cst_rtefind
      inv_find.of_SetRequestor(this)
      Return 1
   END IF
ELSE
...
```

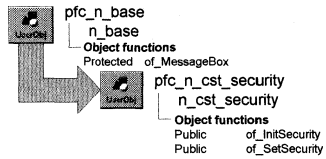
n_cst_security

Description

Security service. PFC can handle many of your application's security needs. It includes administrative components and a runtime security object, n_cst_security. N_cst_security includes instance variables as well as precoded events and functions. This discussion describes only those functions you code in an application.

FOR INFO For complete information on security in PFC, see the *PFC User's Guide*.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_string
n_ds
n_tr

Usage

To use the PFC security system, you must first define users and groups, then associate them with windows and controls, as described in the *PFC User's Guide*.

To use n_cst_security in an application:

1 Create the object by calling the n_cst_appmanager of_SetSecurity function (this example is from an n_cst_appmanager pfc_Open event:

```
this.of_SetSecurity(TRUE)
```

2 Establish a transaction object for the security database. This example assumes an itr_sec instance variable of type n_tr on n_cst_appmanager:

```
itr_sec = CREATE n_tr
CONNECT using itr_sec;
```

3 Initialize the security object by calling the of_InitSecurity function:

```
this.inv_security.of_InitSecurity &
   (itr_sec, "EISAPP", &
      gnv_app.of_GetUserID(), "Default")
```

N_cst_appmanager pfc_Open
The steps listed above might all be coded in the n_cst_appmanager pfc_Open event.

4 Disconnect from the database and destroy the transaction object when the application closes. This example might be coded in the n_cst_appmanager pfc_Close event:

```
DISCONNECT using itr_sec;
Destroy itr_sec
```

5 In the Open or pfc_PreOpen events of windows for which you want to apply security, call the of_SetSecurity function:

```
IF NOT &
   gnv_app.inv_security.of_SetSecurity(this) THEN
      MessageBox("Security", &
         "Unable to set security")
      Close(this)
END IF
```

See also n_cst_appmanager

Functions

N_cst_security includes many precoded object functions. The functions you call from an application are documented here:

of_InitSecurity
of_SetSecurity

of_InitSecurity

Description Initializes security parameters.

Access Public

Syntax *application.instancename*.**of_InitSecurity** (*transaction*, *appname*, *user*, *group* {, *precache* })

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
instancename	Instance name of n_cst_security (the n_cst_appmanager default is inv_security)
transaction	N_tr variable for the security database connection. You must connect to the database before calling this function
appname	String specifying the application name, as defined in the security database
user	String specifying the user ID, as defined in the security database
group	String specifying the default group. PFC uses this group if no group is defined in the database for *user*
precache (optional)	Boolean specifying whether to preload all security settings for the user (the default is FALSE)

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Transaction not connected
- **-2** Application not in database
- **-3** User ID not in database
- **-4** User is not a member of any groups and specified default group does not exist
- **-5** Default group not specified
- **-6** Retrieval error

Usage

You typically call this function in the n_cst_appmanager pfc_Open event.

You must call the n_cst_appmanager of_SetSecurity function before calling this function. You must call this function before calling the n_cst_security of_SetSecurity function.

Examples

This example calls the of_InitSecurity function. It assumes an itr_sec instance variable of type n_tr:

```
Integer  li_return

itr_sec = CREATE n_tr
```

```
CONNECT using itr_sec;

li_return = this.inv_security.of_InitSecurity &
   (this.itr_sec, "EISAPP", &
      gnv_app.of_GetUserID(), "Default")
...
```

of_SetSecurity

Description

Establishes security for a window, menu, or window control, hiding and/or disabling controls, as specified for the user in the security database.

Access

Public

Syntax

application.instancename.**of_SetSecurity** (*object*)

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
instancename	Instance name of n_cst_security (the n_cst_appmanager default for this is inv_security)
object	Window, GraphicObject, or DataWindow variable containing the window, DataWindow, or window control for which security is applied

Return value

Boolean. Returns TRUE if the function succeeds and FALSE if an error occurs.

Usage

Call this function in a window Open or pfc_PreOpen event to apply security to a window, hiding and/or disabling controls, as specified for the user in the security database.

Examples

This example calls the of_SetSecurity function:

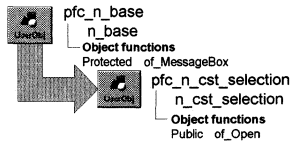
```
IF NOT &
   gnv_app.inv_security.of_SetSecurity(this) THEN
      MessageBox("Security", &
         "Unable to set security")
      Close(this)
END IF
```

n_cst_selection

Description

Selection service. This service provides functions that you can call to display the w_selection dialog box, which allows users to choose a row, which is then passed back to your application.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_tr
n_cst_selectionattrib
w_selection

Usage

To use n_cst_selection functions:

1 Declare a local or instance variable of type n_cst_selection.

2 Establish the information from which the user selects.

3 Call the of_Open function, passing the appropriate arguments.

4 Retrieve the selected row returned by of_Open and continue processing as appropriate.

See also

n_tr

Instance variables

Instance variable	Description	Data type	Access	Usage
is_defaulttitle	Default title for w_selection dialog	String	Protected	Used if you don't specify the title argument

Functions

N_cst_selection includes a precoded function:

of_Open

of_Open

Opens the w_selection dialog box, allowing the user to choose one or more rows for further processing. There are three syntaxes:

To	Use
Open w_selection, displaying retrieved rows from a DataWindow	Syntax 1
Open w_selection, displaying a passed set of rows	Syntax 2
Open w_selection, displaying data stored with the DataWindow object	Syntax 3

Syntax 1

To display retrieved rows for a DataWindow

Description

Displays the w_selection dialog box retrieving rows from the database for the specified DataWindow object. The user selects a one or more columns in w_selection, which returns selection information for the selected columns.

Access

Public

Syntax

instancename.**of_Open** (*dataobject*, *returnval* {, *transobj* {, *columns* {, *arguments* {, *title* } }})

Argument	Description
instancename	Instance name of n_cst_selection
dataobject	String specifying the DataWindow object that w_selection uses to retrieve and display rows
returnval	Array of the Any data type into which the w_selection dialog box places information for the selected columns (passed by reference)
transobj (optional)	N_tr-based Transaction object specifying an active connection that the w_selection dialog box uses to retrieve rows
columns (optional)	String specifying the DataWindow columns that w_selection returns. The default is the first column
arguments (optional)	Twenty-element array of the Any data type specifying retrieval arguments for *dataobject*

Argument	Description
title (optional)	String specifying a title for the w_selection dialog box

Return value

Integer. Returns the number of elements in the *returnval* array if the function succeeds and -1 if an error occurs.

Usage

Call this function when you want the user to choose from a list of values retrieved with the specified DataWindow.

Use Syntax 2 to display passed data in w_selection and Syntax 3 to display data stored in a DataWindow object.

Examples

This example calls the of_Open function:

```
Any   la_values[ ]
String   ls_col[ ]

ls_col[1] = "emp_id"
inv_selection.of_Open("d_emplist", la_values, &
   SQLCA, ls_col)
...
```

Syntax 2

To display a passed set of rows

Description

Displays the w_selection dialog box using rows passed in a function argument. The user selects a one or more columns in w_selection, which returns selection information for the selected columns.

Access

Public

Syntax

instancename.**of_Open** (*dataobject*, *returnval* {, *powerobj* {, *columns* {, *title* } } })

Argument	Description
instancename	Instance name of n_cst_selection
dataobject	String specifying the DataWindow object that w_selection uses to display rows
returnval	Array of the Any data type into which the w_selection dialog box places information for the selected columns (passed by reference)
powerobj *(optional)*	PowerObject array containing data that the w_selection dialog box displays in the *dataobject* DataWindow

Argument	Description
columns (optional)	String specifying the DataWindow columns that w_selection returns. The default is the first column
title (optional)	String specifying a title for the w_selection dialog box

Return value

Integer. Returns the number of elements in the *returnval* array if the function succeeds and -1 if an error occurs.

Usage

Call this function when you want the user to choose from a list of values in a specified DataWindow. Data for the DataWindow is supplied as a function argument.

Use Syntax 1 to display data retrieved from the database and Syntax 3 to display data stored in a DataWindow object.

Examples

This example calls the of_Open function:

```
Any  la_values[ ]
n_cst_selection   lnv_selection
String   ls_colreturn[ ]
Integer   li_return
s_svalue   lstr_svalue[ ]
...
lstr_svalue[1].s_value = "Massachusetts"
lstr_svalue[2].s_value = "Vermont"
lstr_svalue[3].s_value = "New Hampshire"
lstr_svalue[4].s_value = "Maine"
lstr_svalue[5].s_value = "Rhode Island"
lstr_svalue[6].s_value = "Connecticut"
lstr_svalue[7].s_value = "New York"

li_return = lnv_selection.of_Open("d_svalue", &
   la_values, lstr_svalue, ls_colreturn[], &
      "New England States")
...
```

Syntax 3

To display data stored with the DataWindow object

Description

Displays the w_selection dialog box using data stored in the specified DataWindow object. The user selects a one or more columns in w_selection, which returns selection information for the selected columns.

Access

Public

Syntax

instancename.**of_Open** (*dataobject*, *returnval* {, *columns* {, *title* } })

Argument	Description
instancename	Instance name of n_cst_selection
dataobject	String specifying the DataWindow object that w_selection uses to retrieve and display rows
returnval	Array of the Any data type into which the w_selection dialog box places information for the selected columns (passed by reference)
columns (optional)	String specifying the DataWindow columns that w_selection returns. The default is the first column
title (optional)	String specifying a title for the w_selection dialog box

Return value

Integer. Returns the number of elements in the *returnval* array.

Usage

Call this function display the w_selection dialog box using data stored in the specified DataWindow object.

Use Syntax 1 to display data retrieved from the database and Syntax 2 to display passed data.

Examples

This example calls the of_Open function:

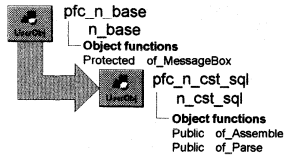
```
Any   ls_values[ ]

inv_selection.of_Open("d_emplist", ls_values)
...
```

n_cst_sql

Description

SQL parsing service. This service provides functions that you can call to assemble and parse SQL statements.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_sqlattrib

Usage

To use n_cst_sql functions:

1 Declare a variable of type n_cst_sql:

```
n_cst_sql    inv_sql
```

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements.

2 Call n_cst_sql functions as needed.

See also

n_ds
u_dw

Functions

N_cst_sql includes precoded object functions:

of_Assemble
of_Parse

of_Assemble

Description

Builds a SQL statement from its component parts.

Access

Public

Syntax

instancename.**of_Assemble** (*components*)

Argument	Description
instancename	Instance name of n_cst_sql
components	Array of n_cst_sqlattrib structures, each element of which contains a SQL statement that the function combines using a UNION statement

Return value

String. Returns the completed SQL statement. An empty string indicates an error.

Usage

Although the *components* argument uses an array, you need to use the first element only if the SQL statement contains no UNION statements.

Examples

This example calls the of_Assemble function. It assumes an inv_sql instance variable:

```
String   ls_sql
n_cst_sqlattrib   lnv_sqlattrib[ ]

lnv_sqlattrib[1].s_verb = sle_verb.text
lnv_sqlattrib[1].s_tables = sle_tables.text
lnv_sqlattrib[1].s_columns = sle_columns.text
lnv_sqlattrib[1].s_values = sle_values.text
lnv_sqlattrib[1].s_where = sle_where.text
lnv_sqlattrib[1].s_order = sle_order.text
lnv_sqlattrib[1].s_group = sle_group.text
lnv_sqlattrib[1].s_having = sle_having.text

ls_sql = inv_sql.of_Assemble &
   (lstr_sql)
MessageBox("SQL", ls_sql)
```

of_Parse

Description

Separates a SQL statement into its component parts.

Access

Public

Syntax

instancename.**of_Parse** (*statement, components*)

Argument	Description
instancename	Instance name of n_cst_sql
statement	String containing the SQL statement to be separated

Argument	Description
components	Array of type n_cst_sqlattrib into which the function places SQL statement components (passed by reference)

Return value

Integer. Returns the number of elements in the *components* array.

Examples

This example calls the of_Parse function. It assumes a single element in the lstr_sql array and an inv_sql instance variable:

```
String   ls_sql
n_cst_sqlattrib   lnv_sqlattrib[ ]
Integer   li_return

li_return = inv_sql.of_Parse &
   (mle_sql.text, lnv_sqlattrib)
IF li_return > 0 THEN
   sle_verb.text= lnv_sqlattrib[1].s_verb
   sle_tables.text = lnv_sqlattrib[1].s_tables
   sle_columns.text = lnv_sqlattrib[1].s_columns
   sle_values.text = lnv_sqlattrib[1].s_values
   sle_where.text = lnv_sqlattrib[1].s_where
   sle_order.text = lnv_sqlattrib[1].s_order
   sle_group.text = lnv_sqlattrib[1].s_group
   sle_having.text= lnv_sqlattrib[1].s_having
END IF
```

n_cst_sqlspy

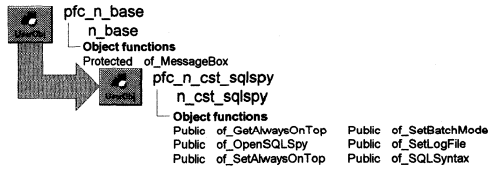
Description

SQL Spy debugging service, a development tool that helps you debug applications by logging and optionally displaying the SQL that DataWindows submit to the database. Do not use this functionality in production applications.

N_cst_sqlspy includes instance variables as well as precoded events and functions. This discussion describes only those functions you code in a typical application.

FOR INFO For complete information on using SQL Spy, see the *PFC User's Guide*.

Ancestry



Library

PFUTIL.PBL
PFEUTIL.PBL

Object relationships

d_sqlspyprint
n_cst_conversion
n_ds
w_sqlspy
w_sqlspyinspect

Usage

You typically call most n_cst_sqlspy functions from within the application manager. To use SQL Spy:

1 Enable the debugging service by calling the n_cst_appmanager of_SetDebug function:

```
this.of_SetDebug(TRUE)
```

2 Enable SQL Spy by calling the n_cst_debug of_SetSQLSpy function:

```
this.inv_debug.of_SetSQLSpy(TRUE)
```

3 Specify the log file by calling the n_cst_sqlspy of_SetLogFile function:

```
this.inv_debug.inv_sqlspy.of_SetLogFile &
   ("c:\debug\temp\appdbug.log")
```

4 Call other n_cst_sqlspy functions, as necessary to log and optionally display DataWindow update SQL.

See also

n_cst_appmanager
n_cst_debug

Functions

N_cst_sqlspy includes many precoded object functions. The functions you call from an application are documented here:

of_GetAlwaysOnTop	of_SetBatchMode
of_OpenSQLSpy	of_SetLogFile
of_SetAlwaysOnTop	of_SQLSyntax

of_GetAlwaysOnTop

Description

Reports whether the w_sqlspy window always displays on top of other windows.

Access

Public

Syntax

*application.debug.sqlspy.***of_GetAlwaysOnTop** ( )

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
debug	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
sqlspy	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)

Return value

Boolean. Returns TRUE if the w_sqlspy window is always on top and FALSE if it is not.

Examples

This example calls the of_GetAlwaysOnTop function:

```
Boolean  ib_ontop

ib_ontop = &
   gnv_app.inv_debug.inv_sqlspy.of_GetAlwaysOnTop()
...
```

of_OpenSQLSpy

Description

Opens or closes the w_sqlspy window. This window displays the contents of the log file specified in the of_SetLogFile function.

Access

Public

Syntax

*application.debug.sqlspy.***of_OpenSQLSpy** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
debug	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
sqlspy	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
boolean	Boolean specifying whether to open (TRUE) or close (FALSE) the w_sqlspy window

Return value

Integer. Returns 1 if the function succeeds, -1 if an error occurs, and 0 if no action is taken.

Examples

This example calls the of_OpenSQLSpy function:

```
gnv_app.inv_debug.inv_sqlspy.of_OpenSQLSpy(TRUE)
```

of_SetAlwaysOnTop

Description

Controls whether the w_sqlspy window always displays on top of other windows.

Access

Public

Syntax

*application.debug.sqlspy.***of_SetAlwaysOnTop** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
debug	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
sqlspy	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)

Argument	Description
boolean	Boolean indicating whether to always display w_sqlspy on top of other windows (TRUE) or to let other windows display on top of w_sqlspy (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetAlwaysOnTop function:

```
...
gnv_app.inv_debug.inv_sqlspy.of_SetAlwaysOnTop &
    (FALSE)
...
```

of_SetBatchMode

Description

Controls whether the service displays the w_sqlspyinspect dialog box before updating the database. This dialog box displays the SQL to be submitted to the database.

Access

Public

Syntax

*application.debug.sqlspy.***of_SetBatchMode** (*boolean*)

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
debug	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
sqlspy	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
boolean	Boolean indicating whether to display the w_sqlspyinspect dialog box (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBatchMode function:

```
gnv_app.inv_debug.inv_sqlspy.of_SetBatchMode &
   (FALSE)
```

of_SetLogFile

Description

Specifies the file into which the SQL Spy service logs DataWindow SQL.

Access

Public

Syntax

*application.debug.sqlspy.***of_SetLogFile** (*logfile*)

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
debug	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
sqlspy	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
logfile	Fully qualified name of the file into which the SQL Spy service copies DataWindow update SQL and other information

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function with an empty string to disable log file processing.

Examples

This example calls the of_SetLogFile function:

```
this.inv_debug.inv_sqlspy.of_SetLogFile &
   ("c:\debug\temp\appdbug.log")
```

of_SQLSyntax

Description

Specifies SQL syntax to be copied to the log file, optionally displaying it in the w_sqlspyinspect dialog box.

Access

Public

Syntax

*application.debug.sqlspy.***of_SQLSyntax** (*heading* {, *sql* {, *oktopause* } })

Argument	Description
application	Global variable of type n_cst_appmanager (this must be gnv_app)
debug	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
sqlspy	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)

Argument	Description
heading	String specifying a heading for the log file
sql (optional)	String specifying the SQL statement to be logged
oktopause (optional)	Boolean specifying whether to pause statement execution for display in the w_sqlspyinspect dialog box. The default is FALSE

Return value

Integer. Returns values as follows:

- 2 Function succeeded; do not process this statement
- 1 Function succeeded; do not process other pending updates
- 0 Function succeeded; continue processing DataWindow updates
- -1 Error

Usage

The SQL Spy service calls this function automatically for all DataWindow and EXEC IMMEDIATE statements automatically. You can call it to log native SQL statements.

To display the statement in the w_sqlspyinspect dialog box, *oktopause* must be TRUE and batch mode must have been disabled previously by calling the of_SetBatchMode(FALSE) function

Examples

This example calls the of_SQLSyntax function:

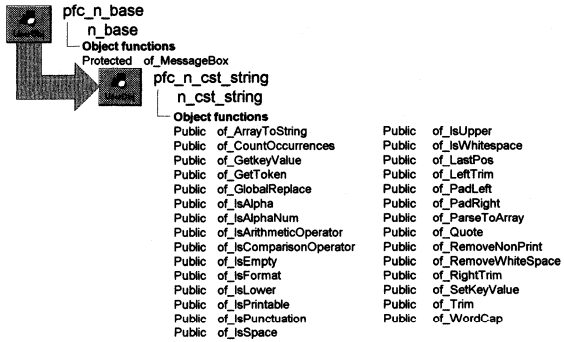
```
String   ls_sql

ls_sql = "SELECT * FROM employee;"
gnv_app.inv_debug.inv_sqlspy.of_SQLSyntax &
   ("Native SQL", ls_sql, TRUE)
...
```

n_cst_string

Description

String parsing service. This service provides functions that you can call to manipulate string data.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_numerical

Usage

To use string functions:

1 Declare a variable of type n_cst_string:

```
n_cst_string   inv_string
```

Because PFC defines this object with the autoinstantiate option, you don't need to code CREATE or DESTROY statements.

2 Call string service functions as needed.

See also

n_cst_conversion

Functions

N_cst_string includes precoded object functions:

of_ArrayToString	of_IsUpper
of_CountOccurrences	of_IsWhiteSpace
of_GetKeyValue	of_LastPos
of_GetToken	of_LeftTrim
of_GlobalReplace	of_PadLeft

of_IsAlpha	of_PadRight
of_IsAlphaNum	of_ParseToArray
of_IsArithmeticOperator	of_Quote
of_IsComparisonOperator	of_RemoveNonPrint
of_IsEmpty	of_RemoveWhiteSpace
of_IsFormat	of_RightTrim
of_IsLower	of_SetKeyValue
of_IsPrintable	of_Trim
of_IsPunctuation	of_WordCap
of_IsSpace	

of_ArrayToString

Description

Creates a single string from an array of strings.

Access

Public

Syntax

instancename.**of_ArrayToString** (*source*, *delimiter*, *target*)

Argument	Description
instancename	Instance name of n_cst_string
source	String array to be converted into a single string
delimiter	String specifying the delimiter the function places between array elements
target	String to contain the *source* array (passed by reference)

Return value

Long. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_ArrayToString function:

```
String   ls_array[ ], ls_single
Long     ll_return
String   ls_delimiter = "~t"

// logic to populate ls_array
...
ll_return = inv_string.of_ArrayToString &
   (ls_array, ls_delimiter, ls_single)
```

of_CountOccurrences

Description

Counts the number of occurrences of one string within another string.

Access

Public

Syntax

instancename.**of_CountOccurrences** (*source*, *target* {, *ignorecase* })

Argument	Description
instancename	Instance name of n_cst_string
source	String to be searched
target	String to search for
ignorecase (optional)	Boolean indicating whether to search for exact upper-lowercase match. The default is TRUE

Return value

Long. Returns the number of occurrences of *target* within *source*.

Examples

This example calls the of_CountOccurrences function:

```
String   ls_source, ls_target
Long  ll_count

ls_source = mle_source.text
ls_target = mle_target.text
ll_count = inv_string.of_CountOccurrences &
   (ls_source, ls_target, TRUE)
MessageBox("String", String(ll_count) &
   + " occurrences of " + ls_target &
   + " within " + ls_source )
```

of_GetKeyValue

Description

Retrieves the right side of a *keyword=value* string.

Access

Public

Syntax

instancename.**of_GetKeyValue** (*source*, *keyword*, *separator*)

Argument	Description
instancename	Instance name of n_cst_string
source	String containing the complete expression
keyword	String containing the keyword to be searched for
separator	String containing the separator character in *source*

Return value

String. Returns the value corresponding to *keyword*. If *keyword* is not in *source*, the function returns an empty string; if any argument's value is NULL, the function returns NULL.

Examples

This example is from the w_sheet pfc_ControlGotFocus event:

```
String   ls_microhelp
n_cst_string   lnv_string
...
ls_microhelp = lnv_string.of_GetKeyValue &
   (idrg_current.tag, "microhelp", ";")
...
```

of_GetToken

Description

Retrieves the token from a passed string. This function receives as arguments a string from which the token is to be removed (from the left) and the separator character.

What is a token?
A token is a collection of characters separated by a delimiter.

If the separator character appears in the string, the function returns the token, not including the separator character. If the separator character does not appear in the string, the function returns the entire string. In either case, the source string is truncated on the left by the length of the token and separator character if any.

Access

Public

Syntax

instancename.**of_GetToken** (*source*, *separator*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be parsed (passed by reference)
separator	String containing the separator

Return value

String. Returns the token if *separator* is found within *source* and returns the entire *source* if *separator* is not found.

Usage

Call this function to break up strings with embedded values into multiple separate strings.

Examples

This example calls the of_GetToken function:

```
String  ls_parm[ ], ls_type, ls_colname, ls_level
Integer  li_args, li_cnt, li_p1, li_p2, li_level
n_cst_string  lnv_string

// Parse the comma-delimited string
li_args = 1
ls_parm[li_args] = &
Trim(lnv_string.of_GetToken(as_args, ","))
DO WHILE Len(as_args) > 0
   li_args ++
   ls_parm[li_args] = &
      Trim(lnv_string.of_GetToken(as_args, ","))
Loop
...
```

of_GlobalReplace

Description

Replaces all occurrences of one string within another string.

Access

Public

Syntax

instancename.**of_GlobalReplace** (*source*, *lookfor*, *replacewith* {, *ignorecase*})

Argument	Description
instancename	Instance name of n_cst_string
source	String to be searched
lookfor	String to be found
replacewith	String used to replace *lookfor*
ignorecase (optional)	Boolean indicating whether to search for exact upper-lowercase match. The default is TRUE

Return value

String. Returns *source* with all occurrences of *lookfor* replaced with *replacewith*.

Examples

This example calls the of_GlobalReplace function:

```
String   ls_source, ls_original
String   ls_replace, ls_new
```

```
ls_source = mle_source.text
ls_original = mle_original.text
ls_replace = mle_replacement.text
ls_new = inv_string.of_GlobalReplace &
   (ls_source, ls_original, ls_replace)
MessageBox("String", "New String is: " &
   + ls_new )
```

of_IsAlpha

Description

Reports whether a passed string contains only alphabetic characters (a-z and A-Z.)

Access

Public

Syntax

instancename.**of_IsAlpha** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only alphabetic characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsAlpha function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsAlpha(ls_source) THEN
   MessageBox("String", ls_source &
      + " contains only alphabetic characters")
ELSE
   MessageBox("String", ls_source &
      + " contains nonalphabetic characters")
END IF
```

of_IsAlphaNum

Description

Reports whether a passed string contains only alphanumeric characters (a-z, A-Z, and 0-9.)

Access

Public

Syntax

instancename.**of_IsAlphaNum** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only alphanumeric characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsAlphaNum function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsAlphaNum(ls_source) THEN
   MessageBox("String", ls_source &
      + " contains only alphanumeric characters")
ELSE
   MessageBox("String", ls_source &
      + " contains nonalphanumeric characters")
END IF
```

of_IsArithmeticOperator

Description

Reports whether a passed string contains only arithmetic operator characters. Arithmetic operator characters are (,), +, -, *, /, and ^.

Access

Public

Syntax

instancename.**of_IsArithmeticOperator** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only arithmetic operator characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsArithmeticOperator function:

```
String   ls_source
```

```
ls_source = mle_source.text
IF inv_string.of_IsArithmeticOperator(ls_source) THEN
   MessageBox("String", ls_source &
      + " contains only arithmetic operators")
...
```

of_IsComparisonOperator

Description

Reports whether a passed string contains only comparison operator characters. Comparison operator characters are <, >, and =.

Access

Public

Syntax

instancename.**of_IsComparisonOperator** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only comparison operator characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsComparisonOperator function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsComparisonOperator(ls_source) &
   THEN
      MessageBox("String", ls_source &
         + " contains only comparison operators")
...
```

of_IsEmpty

Description

Reports whether a passed string is either empty or NULL.

Access

Public

Syntax

instancename.**of_IsEmpty** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* is empty or NULL and FALSE if it is neither.

Examples

This example calls the of_IsEmpty function:

```
String   ls_source

ls_source = sle_directory.text
IF inv_string.of_IsEmpty(ls_source) THEN
   MessageBox("String", &
      "You must specify a directory.")
END IF
```

of_IsFormat

Description

Reports whether a passed string contains only format characters. Format characters are all nonalphanumeric printable characters.

Access

Public

Syntax

instancename.**of_IsFormat** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only format characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsFormat function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsFormat(ls_source) THEN
   MessageBox("String", ls_source &
      + " contains only format characters")
...
```

of_IsLower

Description

Reports whether the alphabetic characters in a passed string are all lowercase.

Access

Public

Syntax

instancename.**of_IsLower** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if the alphabetic characters in *source* are all lowercase and FALSE if they are not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsLower function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsAlpha(ls_source) THEN
   IF inv_string.of_IsLower(ls_source) THEN
      MessageBox("String", ls_source &
         + " contains only lowercase letters")
...
```

of_IsPrintable

Description

Reports whether a passed string contains only printable characters (ASCII 32 through ASCII 126).

Access

Public

Syntax

instancename.**of_IsPrintable** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only printable characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example is from the of_LeftTrim function:

```
Character   lc_char
Boolean     lb_char
...
DO WHILE Len(as_source) > 0 AND NOT lb_char
   lc_char = as_source
      IF of_IsPrintable(lc_char) AND &
         NOT of_IsSpace(lc_char) THEN
            lb_char = TRUE
...
```

of_IsPunctuation

Description

Reports whether a passed string contains only punctuation characters. Punctuation characters are ! " ' . : ; ? and the comma.

Access

Public

Syntax

instancename.**of_IsPunctuation** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only punctuation characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsPunctuation function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsPunctuation(ls_source) THEN
   MessageBox("String", ls_source &
      + " contains only punctuation")
...
```

of_IsSpace

Description

Reports whether a passed string contains only spaces.

Access

Public

Syntax

instancename.**of_IsSpace** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only spaces and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsSpace function:

```
String   ls_source

ls_source = sle_directory.text
IF inv_string.of_IsEmpty(ls_source) OR &
   inv_string.of_IsSpace(ls_source) THEN
      MessageBox("String", &
         "You must specify a directory.")
   Return
END IF
...
```

of_IsUpper

Description

Reports whether the alphabetic characters in a passed string are all uppercase.

Access

Public

Syntax

instancename.**of_IsUpper** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if the alphabetic characters in *source* are all uppercase and FALSE if they are not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsUpper function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsAlpha(ls_source) THEN
   IF inv_string.of_IsUpper(ls_source) THEN
```

```
        MessageBox("String", ls_source &
           + " contains only uppercase letters")
...
```

of_IsWhiteSpace

Description

Reports whether a passed string contains only whitespace characters (newline, tab, vertical tab, carriage return, form feed, backspace, and space).

Access

Public

Syntax

instancename.**of_IsWhiteSpace** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be tested

Return value

Boolean. Returns TRUE if *source* contains only whitespace characters and FALSE if it does not. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_IsWhiteSpace function:

```
String   ls_source

ls_source = mle_source.text
IF inv_string.of_IsWhiteSpace(ls_source) THEN
   MessageBox("String", ls_source &
      + " contains only whitespace")
...
```

of_LastPos

Description

Searches backward through a string to find the last occurrence of another string.

Access

Public

Syntax

instancename.**of_LastPos** (*source*, *target* {, *start* })

Argument	Description
instancename	Instance name of n_cst_string
source	String containing the complete expression

Argument	Description
target	String containing the expression to be searched for
start (optional)	Long specifying the starting position. The default is 0 (start with the last character)

Return value

Long. Returns the position of the leftmost character of the last occurrence of *target* if it is found and 0 if *target* is not found.

Usage

This function complements the PowerScript Pos function, which finds the first occurrence of one string within another string.

Examples

This example is from the n_cst_filesrv of_ParsePath function:

```
Integer  li_pos
String   ls_file
n_cst_string  lnv_string
...
// Get the file name and extension.
li_pos = lnv_string.of_LastPos &
   (as_path, is_separator, 0)
ls_file = Right(as_path, (Len(as_path) - li_pos))
as_path = Left(as_path, li_pos)
...
```

of_LeftTrim

Description

Removes spaces and nonprintable characters from the beginning of a string.

Access

Public

Syntax

instancename.**of_LeftTrim** (*source* {, *removespaces* {, *removenonprintable* } })

Argument	Description
instancename	Instance name of n_cst_string
source	String to be trimmed
removespaces (optional)	Boolean specifying whether to remove spaces. The default is TRUE
removenonprintable (optional)	Boolean specifying whether to remove nonprintable characters (for example, newline, tab, and carriage return). The default is FALSE

Return value

String. Returns *source* with the specified characters removed. If any argument's value is NULL, the function returns NULL.

Usage

This function complements the PowerScript Trim function, which removes both leading and trailing spaces from a string.

Examples

This example calls the of_LeftTrim function:

```
...
IF ll_first > 0 and ll_last > 0 THEN
   ls_line = lnv_string.of_LeftTrim &
      (ls_line, TRUE, TRUE)
   IF Left (ls_line, 1) = "[" THEN
      ll_pos = Pos (ls_line, "]")
...
```

of_PadLeft

Description

Pads a string with spaces on the left until it is a specified length.

Access

Public

Syntax

instancename.**of_PadLeft** (*source*, *length*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be expanded
length	Long specifying the new length

Return value

String. Returns *source* with left-padded spaces to make it a *length* characters long. If any argument's value is NULL, the function returns NULL.

Examples

This example calls the of_PadLeft function:

```
String   ls_source, ls_new

ls_source = mle_source.text
ls_new = inv_string.of_PadLeft(ls_source, 40)
MessageBox("String", "Original is: '" &
   + ls_source + "' ~r~nPadded string is: '" &
   + ls_new + "'" )
```

of_PadRight

Description

Pads a string with spaces on the right until it is a specified length.

Access

Public

Syntax

instancename.**of_PadRight** (*source*, *length*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be expanded
length	Long specifying the new length

Return value

String. Returns *source* with right-padded spaces to make it a *length* characters long. If any argument's value is NULL, the function returns NULL.

Examples

This example calls the of_PadRight function:

```
String   ls_source, ls_new

ls_source = mle_source.text
ls_new = inv_string.of_PadRight(ls_source, 40)
MessageBox("String", "Original is: '" &
   + ls_source + "' ~r~nPadded string is: '" &
   + ls_new + "'" )
```

of_ParseToArray

Description

Parses a string into array elements, based on a delimiter string.

Access

Public

Syntax

instancename.**of_ParseToArray** (*source*, *delimiter*, *array*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be converted into an array
delimiter	String specifying the character(s) that separate array elements in *source*
array	String array into which the function places the extracted strings (passed by reference)

Return value

Long. Returns the number of elements in *array*. If *source* or *delimiter* are NULL, the function returns NULL.

Examples

This example is from the n_cst_sql of_Parse function:

```
Integer  li_len, li_pos, li_kwnum, li_numstats
Integer  li_cnt
String   ls_uppersql, ls_keyword[7], ls_clause[7]
String   ls_sql[ ]
n_cst_string  lnv_string

// Separate the statement into multiple statements,
// Separated by UNIONs
li_numstats = lnv_string.of_Parsetoarray &
   (as_sql, "UNION", ls_sql)
...
```

of_Quote

Description

Encloses a string in double quotes.

Access

Public

Syntax

instancename.**of_Quote** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String to be enclosed in double quotes

Return value

String. Returns *source* enclosed in double quotes.

Examples

This example calls the of_Quote function:

```
String  ls_source, ls_new

ls_source = mle_source.text
ls_new = inv_string.of_Quote(ls_source)
MessageBox("String", "Original is: " &
   + ls_source + "~r~nNew String is: " &
   + ls_new)
```

of_RemoveNonPrint

Description

Removes all nonprintable characters from a string. Nonprintable characters are:

ASCII 31 and lower
ASCII 127 and higher

Access

Public

Syntax

instancename.**of_RemoveNonPrint** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String from which the function removes nonprintable characters

Return value

String. Returns *source* with nonprintable characters removed. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_RemoveNonPrint function:

```
...
inv_string.of_RemoveNonPrint &
   (ls_original[li_count])
...
```

of_RemoveWhiteSpace

Description

Removes all whitespace characters from a string. Whitespace characters are newline, tab, vertical tab, carriage return, form feed, backspace, and space.

Access

Public

Syntax

instancename.**of_RemoveWhiteSpace** (*source*)

Argument	Description
instancename	Instance name of n_cst_string
source	String from which the function removes whitespace characters

Return value

String. Returns *source* with whitespace characters removed. If *source* is NULL, the function returns NULL.

Examples

This example calls the of_RemoveWhiteSpace function:

```
String   ls_original[ ]
```

```
Integer  li_return, li_count

li_return = inv_filesrv.of_FileRead &
   (sle_file.text, ls_original)
CHOOSE CASE li_return
   CASE -1
      MessageBox("Error", "Error accessing file")
   CASE ELSE
      FOR li_count = 1 to li_return
         mle_target.text = mle_target.text + &
            inv_string.of_RemoveWhiteSpace &
               (ls_original[li_count])
      NEXT
END CHOOSE
```

of_RightTrim

Description Removes spaces and nonprintable characters from the end of a string.

Access Public

Syntax *instancename*.**of_RightTrim** (*source* {, *removespaces* {, *removenonprintable* } })

Argument	Description
instancename	Instance name of n_cst_string
source	String to be trimmed
removespaces (optional)	Boolean specifying whether to remove spaces. The default is TRUE
removenonprintable (optional)	Boolean specifying whether to remove nonprintable characters (for example, newline, tab, and carriage return). The default is FALSE

Return value String. Returns *source* with the specified characters removed. If any argument's value is NULL, the function returns NULL.

Usage This function complements the PowerScript Trim function, which removes both leading and trailing spaces from a string.

Examples This example is from the n_cst_string of_Trim function:

```
...
IF ab_remove_spaces and ab_remove_nonprint THEN
```

```
// Remove spaces and nonprintable characters.
   as_source = of_LeftTrim(as_source, &
      ab_remove_spaces, ab_remove_nonprint)
   as_source = of_RightTrim(as_source, &
...
```

of_SetKeyValue

Description

Sets the value portion of a *keyword=value* expression.

Access

Public

Syntax

instancename.**of_SetKeyValue** (*source, keyword, newvalue, separator*)

Argument	Description
instancename	Instance name of n_cst_string
source	String whose value is to be set (passed by reference)
keyword	String specifying the keyword whose value will be set
newvalue	String containing the new value
separator	String specifying the separator character in *source*

Return value

Long. Returns 1 if the function succeeds, -1 if an error occurs, and NULL if any argument's value is NULL.

Examples

This example calls the of_SetKeyValue function:

```
...
ls_key = inv_string.of_GetToken(ls_temp, "=")
li_return = inv_string.of_SetKeyValue &
   (ls_source, ls_key, ls_value, "=")
...
```

of_Trim

Description

Removes spaces and nonprintable characters from the left and right end of a string.

Access

Public

Syntax

instancename.**of_Trim** (*source* {, *removespaces* {, *removenonprintable* } })

Argument	Description
instancename	Instance name of n_cst_string
source	String to be trimmed
removespaces (optional)	Boolean specifying whether to remove spaces. The default is TRUE
removenonprintable (optional)	Boolean specifying whether to remove nonprintable characters (for example, newline, tab, and carriage return). The default is FALSE

Return value

String. Returns *source* with the specified characters removed. If any argument's value is NULL, the function returns NULL.

Usage

This function complements the PowerScript Trim function, which does not remove nonprintable characters.

Examples

This example calls the of_Trim function:

```
String   ls_key, ls_line
n_cst_string   lnv_string
Integer  li_keys
Long     ll_equal
...
ls_key = lnv_string.of_Trim &
   (Left (ls_line, ll_equal - 1), TRUE, TRUE)
IF Len (ls_key) > 0 THEN
   li_keys++
      as_keys[li_keys] = ls_key
END IF
...
```

of_WordCap

Description

Sets the first letter of each word in a string to uppercase and all remaining letters to lowercase.

Access

Public

Syntax

instancename.**of_WordCap** (*source*)

Argument	Description
instancename	Instance name of n_cst_string

Argument	Description
source	String to be converted

Return value

String. Returns *source* with all words set to initial cap.

Examples

This example calls the of_WordCap function:

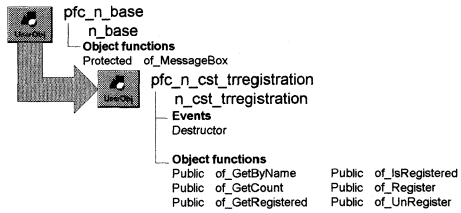
```
String   ls_source, ls_new

ls_source = mle_source.text
ls_new = inv_string.of_WordCap(ls_source)
mle_target.text = ls_new
```

n_cst_trregistration

Description

Transaction registration service.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_tr

Usage

Use this service to keep track of transactions when your application uses more than one transaction. To use this service:

1 Register transactions by calling the of_Register function:

```
gnv_app.inv_trregistration.of_Register &
   (itr_security)
```

2 Provide and track transaction names through the n_tr of_SetName function:

```
itr_security.of_SetName("Security")
```

3 Call n_cst_trregistration functions as needed to access information on registered transactions.

4 (Optional) Unregister transactions by calling the of_UnRegister function:

```
gnv_app.inv_trregistration.of_UnRegister &
   (itr_security)
```

When your application closes, this object automatically destroys all open transactions. Use the n_tr of_SetAutoRollbackction to specify whether transactions are rolled back or committed before being destroyed.

This service is for use with Transaction objects that inherit from n_tr.

Structures

Structure	Field	Data type
os_transentry	b_used	Boolean

Structure	Field	Data type
	tr_object	n_tr

See also

n_cst_appmanager
n_tr

Instance variables

Instance variable	Description	Data type	Access	Usage
istr_trans[ ]	Structure array that tracks Transaction objects and whether the array slot is used	os_transentry	Protected	Used to track Transaction objects
ii_count	Number of current transactions registered with the service	Integer	Protected	Used to track the number of registered Transaction objects. This value is not always the same as the upperbound of the istr_trans array

Events

N_cst_trregistration includes a precoded event:

Destructor

Destructor

Description

Destroys all registered Transaction objects.

Usage

This event ensures that transactions are destroyed when an application terminates. However, it's best if the application does this explicitly.

Functions

N_cst_trregistration includes precoded object functions:

of_GetByName	of_IsRegistered
of_GetCount	of_Register
of_GetRegistered	of_UnRegister

of_GetByName

Description

Retrieves a reference to a registered Transaction object, given its name.

Access

Public

Syntax

appname.instancename.**of_GetByName** (*transname*, *transobject*)

Argument	Description
appname	Instance name of the n_cst_appmanager-based Application object (this must be gnv_app)
instancename	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
transname	String specifying the name of the Transaction object to retrieve
transobject	N_tr variable to contain the reference to the Transaction object (passed by reference)

Return value

Integer. Returns 1 if the specified name is a valid registered Transaction object and 0 if it is not.

Usage

One use of this function is to pass the name between windows so they can use the same Transaction object.

Examples

This example calls the of_GetByName function in a response window's Open event to use a transaction created by another window. The calling window passes the name via the OpenWithParm function:

```
Integer  li_return
String   ls_name
n_tr  lt_trans

ls_name = Message.StringParm
IF ls_name = "" THEN
   MessageBox("Error","No transaction passed")
   CloseWithReturn(this, -1)
END IF
```

```
li_return = &
   gnv_app.inv_trregistration.of_GetByName &
      (ls_name, lt_trans)
IF li_return = 0 THEN
   MessageBox("Error","Transaction not found")
   CloseWithReturn(this, -1)
ELSE
   dw_list.SetTransObject(lt_trans)
END IF
```

of_GetCount

Description

Retrieves the number of registered Transaction objects.

Access

Public

Syntax

appname.instancename.**of_GetCount** ( )

Argument	Description
appname	Instance name of the n_cst_appmanager-based Application object (this must be gnv_app)
instancename	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)

Return value

Integer. Returns the number of registered Transaction objects.

Examples

This example calls the of_GetCount function:

```
Integer  li_count

li_count = gnv_app.inv_trregistration.of_GetCount( )
```

of_GetRegistered

Description

Retrieves all registered Transaction objects.

Access

Public

Syntax

appname.instancename.**of_GetRegistered** (*transobj*)

Argument	Description
appname	Instance name of the n_cst_appmanager-based Application object (this must be gnv_app)

Argument	Description
instancename	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
transobj	N_tr array into which the function places references to all registered Transaction objects (passed by reference)

Return value

Integer. Returns the number of elements in the *transobj* array.

Examples

This example calls the of_GetRegistered function:

```
Integer  li_return
n_tr  ltr_trans[ ]

li_return = &
   gnv_app.inv_trregistration.of_GetRegistered &
      (ltr_trans)
...
```

of_IsRegistered

Description

Reports whether a Transaction object is registered with n_cst_trregistration.

Access

Public

Syntax

appname.instancename.**of_IsRegistered** (*nameortrans*)

Argument	Description
appname	Instance name of the n_cst_appmanager-based Application object (this must be gnv_app)
instancename	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
nameortrans	Either of the following: ♦ String containing the transaction name, as specified in the n_tr of_SetName function ♦ N_tr variable specifying the Transaction object

Return value

Boolean. Returns TRUE if *nameortrans* is registered and FALSE if it is not.

Examples

This example calls the of_IsRegistered function:

```
IF gnv_app.inv_trregistration.of_IsRegistered &
   (SQLCA) THEN
```

```
        MessageBox("Registration", "Registration on")
ELSE
    MessageBox("Registration", "Registration off")
END IF
```

of_Register

Description

Registers a Transaction object.

Access

Public

Syntax

*appname.instancename.***of_Register** (*transobject*)

Argument	Description
appname	Instance name of the n_cst_appmanager-based Application object (this must be gnv_app)
instancename	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
transobject	N_tr variable specifying the Transaction object to register. This Transaction object must already be created

Return value

Integer. Returns 1 if the function succeeds, 0 if the transaction is already registered, and -1 if an error occurs.

Examples

This example calls the of_Register function to register SQLCA.

```
IF SQLCA.of_Connect() <> 0 THEN
    MessageBox("Error","Connect error -- SQLCA")
    HALT
ELSE
    gnv_app.inv_trregistration.of_Register(SQLCA)
END IF
```

of_UnRegister

Description

Removes the specified Transaction object from the list of registered Transaction objects.

Access

Public

Syntax

*appname.instancename.***of_UnRegister** (*transobject*)

Argument	Description
appname	Instance name of the n_cst_appmanager-based Application object (this must be gnv_app)
instancename	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
transobject	N_tr-based transaction variable to unregister

Return value

Integer. Returns 1 if the Transaction object was removed from the list, 0 if the Transaction object was not found in the list of registered Transaction objects, and -1 if an error occurs.

Usage

The n_tr Destructor event calls this function to remove a transaction from the list of registered transactions.

Examples

This example calls the of_UnRegister function:

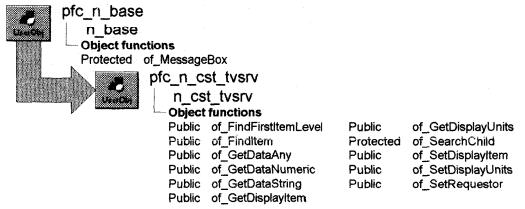
```
...
gnv_app.inv_trregistration.of_UnRegister(itr_sec)
...
```

n_cst_tvsrv

Description

TreeView base service.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

u_tvs

Usage

To use this service:

1 Enable the service by calling the u_tvs of_SetBase function:

```
tv_1.of_SetBase(TRUE)
```

2 Call n_cst_tvsrv functions as needed.

Descendants

n_cst_tvsrv_levelsource
n_cst_tvsrv_print

See also

n_cst_lvsrv

Instance variables

Instance variable	Description	Data type	Access	Usage
is_displayitem	Controls message text displayed for a single row	String	Protected	Call of_SetDisplayItem and of_GetDisplayItem to access
is_displayunits	Controls message text displayed for multiple rows	String	Protected	Call of_SetDisplayUnits and of_GetDisplayUnits to access
itv_requestor	TreeView control that owns the n_cst_tvsrv instance	u_tvs	Protected	Used to access and control the associated TreeView control

Functions

N_cst_tvsrv includes precoded functions:

of_FindFirstItemLevel	of_GetDisplayUnits
of_FindItem	of_SearchChild
of_GetDataAny	of_SetDisplayItem
of_GetDataNumeric	of_SetDisplayUnits
of_GetDataString	of_SetRequestor
of_GetDisplayItem	

of_FindFirstItemLevel

Description

Retrieves the first item on a level.

Access

Public

Syntax

tvcontrol.instancename.**of_FindFirstItemLevel** (*level, handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
level	Integer specifying the TreeView level for which the first item is returned
handle	Long containing the handle of the TreeView item at which to begin searching. Specify 0 to retrieve the first item in the next lower level

Return value

Long. Returns the handle of the first item on the level if the function succeeds, 0 if the item is not found, and -1 if an error occurs.

Examples

This example calls the of_FindFirstItemLevel function:

```
...
// Retrieve level one on the treeview.
li_rc = tv_1.event pfc_populateitem(0)
// Populate the dual datasource for level one.
ll_itemhandle = &
   tv_1.inv_base.of_FindFirstItemLevel(1, 0)
// Populate level two of the treeview,
//  and start the window with the first
// item on level two selected.
tv_1.SelectItem(ll_itemhandle)
tv_1.ExpandItem(ll_itemhandle)
```

```
ll_leveltwoitemhandle = &
   tv_1.inv_base.of_FindFirstItemLevel(2, 0)
tv_1.SelectItem(ll_leveltwoitemhandle)
```

of_FindItem

Description

Searches by label or data for a specified TreeView item.

Access

Public

Syntax

tvcontrol.instancename.**of_FindItem** (*attribute*, *target* {, *beginhandle* {, *level* {, *respectcase* {, *fullcompare* } } } })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
attribute	String specifying the TreeView attribute to search. Valid values are: ♦ Label ♦ Data
target	Any variable specifying the label or data that you are searching for
beginhandle (optional)	Long specifying the handle of the TreeView item from which to begin the search. The default is 0, which starts at the top of the TreeView hierarchy
level (optional)	Integer specifying the level to search. Specify 0 to search all levels
respectcase (optional)	Boolean indicating whether the search is case sensitive (TRUE) or not (FALSE). The default is FALSE
fullcompare (optional)	Boolean indicating whether the specified attribute must match *target* exactly (TRUE) or the attribute can contain *target*. This argument applies only when *target* is a string. The default is FALSE

Return value

Long. Returns the handle of the matching item if the function succeeds, 0 if there are no matching items, and -1 if an error occurs.

Usage

This function calls the u_tvs pfc_SearchCompare event to search either the label or Data property. Override pfc_SearchCompare if you want to perform some other type of comparison.

Levels must have been opened
This function does not find items in levels that have not been opened.

Examples

This example calls the of_FindItem function:

```
Long   ll_return

ll_return = tv_test.inv_base.of_FindItem("label", &
   sle_item.text, 0, 0, FALSE, FALSE)
IF ll_return = -1 THEN
   MessageBox("TreeView", "Find error")
ELSEIF ll_return = 0 THEN
   MessageBox("TreeView", "No matches found")
...
```

of_GetDataAny

Description

Retrieves the specified item's Data property cast to the Any data type.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDataAny** (*handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
handle	Long specifying the handle of the item for which the Data property is returned

Return value

Any. Returns the Data property of the specified TreeView item if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetDataAny function:

```
Long  ll_currenttvitem
Any  la_data

ll_currenttvitem = tv_1.FindItem(currenttreeitem!, 0)
la_data = tv_1.inv_levelsource.of_GetDataAny  &
  (ll_currenttvitem)
MessageBox("TV Data", String(la_data))
```

of_GetDataNumeric

Description

Retrieves the specified item's Data property cast to a number.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDataNumeric** (*handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
handle	Long specifying the handle of the item for which the Data property is returned

Return value

Long. Returns the Data property of the specified TreeView item if the function succeeds and -1 if an error occurs.

Usage

Call this function when the Data property always contains a number.

Examples

This example calls the of_GetDataNumeric function:

```
Long  ll_currenttvitem
Long  ll_data

ll_currenttvitem = tv_1.FindItem(currenttreeitem!, 0)
ll_data = tv_1.inv_levelsource.of_GetDataNumeric  &
  (ll_currenttvitem)
MessageBox("TV Data", String(ll_data))
```

of_GetDataString

Description

Retrieves the specified item's Data property cast to a string

Access

Public

Syntax

tvcontrol.instancename.**of_GetDataString** (*handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
handle	Long specifying the handle of the item for which the Data property is returned

Return value

String. Returns the Data property of the specified TreeView item if the function succeeds and an empty string if an error occurs.

Examples

This example calls the of_GetDataString function:

```
Long  ll_currenttvitem
String  ls_data

ll_currenttvitem = tv_1.FindItem(currenttreeitem!, 0)
ls_data = tv_1.inv_levelsource.of_GetDataString  &
  (ll_currenttvitem)
MessageBox("TV Data", ls_data)
```

of_GetDisplayItem

Description

Retrieves the display name for single items in the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDisplayItem** ( )

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)

Return value

String. Returns the display name for single items in the TreeView.

Usage

PFC uses this name in message text when confirming single-item deletion. For example, if this name is *this row*, PFC displays Are you sure you want to delete this row?

Examples

This example calls the of_GetDisplayItem function:

```
String    ls_item

ls_item = tv_1.inv_base.of_GetDisplayItem( )
MessageBox("TreeView", "Item label is: " &
    + ls_item)
```

of_GetDisplayUnits

Description

Retrieves the display name for multiple rows in the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDisplayUnits** ( )

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)

Return value

String. Returns the display name for multiple items in the TreeView.

Usage

PFC uses this name in message text when confirming multiple item deletion. For example, if this name is *rows*, PFC displays Are you sure you want to delete these rows?

Examples

This example calls the of_GetDisplayUnits function:

```
String    ls_units

ls_units = tv_1.inv_base.of_GetDisplayUnits( )
MessageBox("TreeView", "Unit label is: " &
   + ls_units)
```

of_SearchChild

Description

Searches for a string in either the label or data TreeView item attribute.

Access

Protected

Syntax

tvcontrol.instancename.**of_SearchChild** (*attribute, target, beginhandle, level respectcase, fullcompare*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
attribute	String specifying the TreeView attribute to search. Valid values are: ♦ Label ♦ Data
target	Any variable specifying the label or data to search for
beginhandle	Long specifying the handle of the TreeView item from which to begin the search

Argument	Description
level	Integer specifying the level to search
respectcase	Boolean indicating whether the search is case sensitive (TRUE) or not (FALSE)
fullcompare	Boolean indicating whether the specified attribute must match *target* exactly (TRUE) or the attribute can contain *target*

Return value

Long. Returns the handle of the matching item if the function succeeds, 0 if there are no matching items, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_FindItem function:

```
...
ll_Found = this.of_SearchChild(as_Attribute,  &
   aa_Target, al_Begin, ai_Level, &
      ab_RespectCase, ab_FullCompare)
...
```

of_SetDisplayItem

Description

Specifies the display name for single items in the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_SetDisplayItem** (*itemname*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
itemname	String specifying the display name for single items in the TreeView

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

PFC uses this name in message text when confirming single-row deletion. For example, if you specify *this row*, PFC displays Are you sure you want to delete this row?

Examples

This example calls the of_SetDisplayItem function:

```
...
tv_1.inv_base.of_SetDisplayItem("the selected row")
...
```

of_SetDisplayUnits

Description

Specifies the display name for multiple items in the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_SetDisplayUnits** (*units*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)
units	String specifying the display name for multiple rows of the TreeView

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

PFC uses this name in message text when confirming multiple item deletion. For example, if this name is *rows*, PFC displays Are you sure you want to delete these rows?

Examples

This example calls the of_SetDisplayUnits function:

```
...
tv_1.inv_base.of_SetDisplayUnits("selected rows")
...
```

of_SetRequestor

Description

Associates the requesting object with this instance of n_cst_tvsrv.

Access

Public

Syntax

tvcontrol.instancename.**of_SetRequestor** (*requestor*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv (the u_tvs default is inv_base)

Argument	Description
requestor	U_tvs-based TreeView control to be associated with the service

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_tvs_of_SetBase function:

```
...
if ab_Switch then
  if IsNull(inv_base) Or not IsValid (inv_base) then
    inv_base = create n_cst_tvsrv
    inv_base.of_SetRequestor(this)
    li_rc = 1
  end if
else
...
```

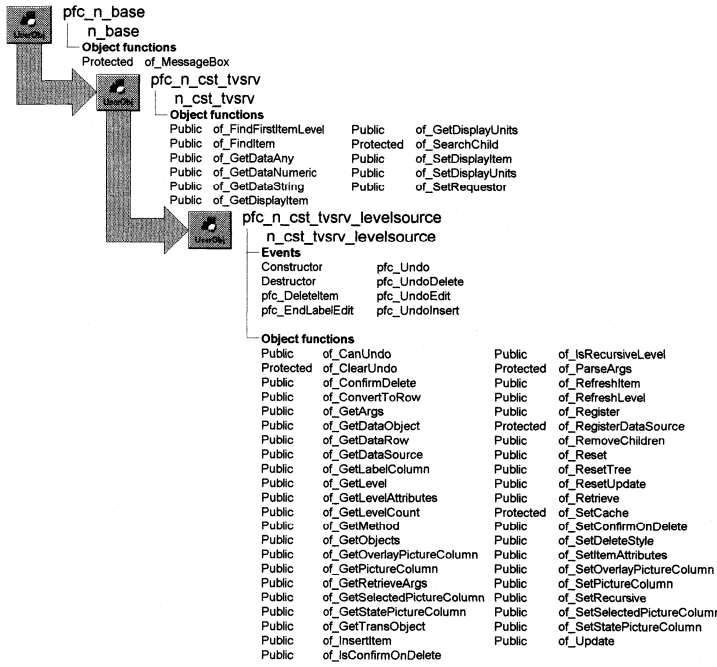
n_cst_tvsrv_levelsource

Description

TreeView data access object. Use this object to display data in a u_tvs control and perform actions on that data. This data can come from a:

SQL statement
DataWindow object
DataStore
DataWindow control
PowerObject

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_dwcache
n_cst_tvsrvattrib
n_ds
n_tr

Usage

To use this object:

1 Enable the service using the u_tvs of_SetDataSource function:

```
this.of_SetDataSource(TRUE)
```

2 Define a data source for each TreeView level by calling the of_Register function, once for each level:

```
this.inv_levelsource.of_Register(1,  &
   "dept_name", "", "d_deptlist", SQLCA, "")
this.inv_levelsource.of_Register(2, "emp_lname",  &
  ":parent.1.dept_id", "d_empbydept", SQLCA, "")
```

3 Call additional functions as needed to control TreeView behavior:

```
this.inv_levelsource.of_SetPictureColumn(1, "1")
this.inv_levelsource.of_SetSelectedPictureColumn  &
   (1, "2")
this.inv_levelsource.of_SetPictureColumn(2, "4")
this.inv_levelsource.of_SetSelectedPictureColumn  &
   (2, "5")
```

4 Populate the TreeView by calling the u_tvs pfc_Populate event:

```
this.event pfc_Populate(0)
```

See also

n_cst_lvsrv_datasource
n_cst_tvsrv_print
u_tvs

Instance variables

Instance variable	Description	Data type	Access	Usage
CACHE_ID	Constant set to "level"	String	Public	Internal
CONTINUE_ACTION	Constant set to 1	Integer	Public	Internal
DELETE_ROWS	Constant set to 1	Integer	Public	Internal
DISCARD_ROWS	Constant set to 0	Integer	Public	Internal
ib_confirmondelete	Indicates whether confirm on delete is enabled	Boolean	Protected	Set with of_SetConfirmOnDelete (default is FALSE)
ib_isrefreshing	Indicates whether a refresh is occurring	Boolean	Protected	Internal

Instance variable	Description	Data type	Access	Usage
ib_isresetting	Indicates whether a reset is occurring	Boolean	Protected	Internal
ib_isundoing	Indicates whether an undo is occurring	Boolean	Protected	Internal
il_undodeletehandle	Deleted item	Long	Protected	Internal
il_undodeleteparenthandle	Parent of deleted item	Long	Protected	Internal
il_undoedithandle	Modified items	Long	Protected	Internal
il_undoinserthandle	Last inserted item	Long	Protected	Internal
inv_attrib[ ]	Data source information	n_cst_tvsrvattrib	Protected	Set with of_Register
inv_cache	Cached data	n_cst_dwcache	Protected	Internal
is_undoeditlabeltext	Modified label text	String	Protected	Internal
is_undotype	Undo type	String	Protected	Internal
itvi_undodeleteitem	Deleted item	TreeViewItem	Protected	Internal
UNDO_DELETE	Constant set to "Delete"	String	Public	Internal
UNDO_EDIT	Constant set to "Edit"	String	Public	Internal
UNDO_INSERT	Constant set to "Insert"	String	Public	Internal

Events

N_cst_tvsrv_levelsource includes precoded events:

Constructor	pfc_Undo
Destructor	pfc_UndoDelete
pfc_DeleteItem	pfc_UndoEdit
pfc_EndLabelEdit	pfc_UndoInsert

Constructor

Description

Enables the caching service.

Usage

This event executes when the object is created.

Destructor

Description

Destroys the caching service.

Usage

This event executes when the object is created.

pfc_DeleteItem

Description

Deletes the item from the TreeView and the row from the DataStore.

Syntax

*tvcontrol.instancename.*EVENT **pfc_DeleteItem** (*handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)
handle	Long specifying the handle of the TreeView item being deleted

Return value

Integer. Returns 1 if the event succeeds and 0 if no action is taken.

Usage

This event does not perform processing if refreshing, resetting, or undoing an insert.

Examples

This example is from the u_tvs pfc_DeleteItem event:

```
...
If isvalid(inv_levelsource) then
   Return inv_levelsource.event pfc_deleteitem  &
      (ll_currentitem)
else
   Return this.deleteitem(ll_currentitem)
end if
...
```

pfc_EndLabelEdit

Description

Changes the label column in the associated data source.

Syntax

*tvcontrol.instancename.*EVENT **pfc_EndLabelEdit** (*handle, newlabel*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control

Argument	Description
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)
handle	Long specifying the handled of the modified TreeView item
newlabel	String specifying the udpated label

Return value

Integer. Returns 0 if the label update is allowed and 1 if PowerBuilder should prevent the label update.

Usage

PFC allows label update only when the associated label column is updatable. If the label column is a computed column, you must override this event to update the computed column and return a 1.

This event does not update the database.

Examples

This example is from the u_tvs EndLabelEdit event:

```
...
if isvalid(inv_levelsource) then
   li_rc = inv_levelsource.event pfc_EndLabelEdit  &
      (handle, newtext)
   // prevent new text from becoming the label
   If li_rc < 1 then Return 1
end if
...
```

pfc_Undo

Description

Restores the last TreeView change.

Syntax

*tvcontrol.instancename.*EVENT **pfc_Undo** ( )

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)

Return value

Integer. Returns 1 if the event succeeds, 0 if there is nothing to undo, and -1 if an error occurs.

Examples

This example is from the u_tvs pfc_Undo event:

```
...
```

```
if isvalid(inv_levelsource) then
   li_rc = inv_levelsource.event pfc_undo()
end if
...
```

pfc_UndoDelete

Description

Restores the last deleted item. This event does not restore children of the last deleted item.

Syntax

tvcontrol.instancename.EVENT **pfc_UndoDelete ( )**

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)

Return value

Integer. Returns 1 if the event succeeds, 0 if there is nothing to undo, and -1 if an error occurs.

Examples

This example is from the pfc_Undo event:

```
...
If of_CanUndo(ls_undotype) then
   Choose Case ls_UndoType
      Case UNDO_DELETE
         return this.event pfc_undodelete()
...
```

pfc_UndoEdit

Description

Restores a label's previous value in the associated DataStore.

Syntax

tvcontrol.instancename.EVENT **pfc_UndoEdit ( )**

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)

Return value

Integer. Returns 1 if the event succeeds, 0 if there is nothing to undo, and -1 if an error occurs.

Usage

PFC allows label update only when the associated label column is updatable. If the label column is a computed column, you must override this event to update the computed column and return a 1.

Examples

This example is from the pfc_Undo event:

```
...
If of_CanUndo(ls_undotype) then
   Choose Case ls_UndoType
      Case UNDO_DELETE
         return this.event pfc_undodelete()
      Case UNDO_INSERT
         return this.event pfc_undoinsert()
      Case UNDO_EDIT
         return this.event pfc_undoedit()
...
```

pfc_UndoInsert

Description

Removes the last inserted item.

Syntax

tvcontrol.instancename.EVENT **pfc_UndoInsert ( )**

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Examples

This example is from the pfc_Undo event:

```
...
Choose Case ls_UndoType
   Case UNDO_DELETE
      return this.event pfc_undodelete()
   Case UNDO_INSERT
      return this.event pfc_undoinsert()
...
```

Functions

N_cst_tvsrv_levelsource includes precoded functions:

of_CanUndo	of_IsRecursiveLevel
of_ClearUndo	of_ParseArgs
of_ConfirmDelete	of_RefreshItem
of_ConvertToRow	of_RefreshLevel
of_GetArgs	of_Register
of_GetDataObject	of_RegisterDataSource
of_GetDataRow	of_RemoveChildren
of_GetDataSource	of_Reset
of_GetLabelColumn	of_ResetTree
of_GetLevel	of_ResetUpdate
of_GetLevelAttributes	of_Retrieve
of_GetLevelCount	of_SetCache
of_GetMethod	of_SetConfirmOnDelete
of_GetObjects	of_SetDeleteStyle
of_GetOverlayPictureColumn	of_SetItemAttributes
of_GetPictureColumn	of_SetOverlayPictureColumn
of_GetRetrieveArgs	of_SetRecursive
of_GetSelectedPictureColumn	of_SetSelectedPictureColumn
of_GetStatePictureColumn	of_SetStatePictureColumn
of_GetTransObject	of_UnRegister
of_InsertItem	of_Update
of_IsConfirmOnDelete	

of_CanUndo

Description

Reports whether the TreeView can undo the last edit, insertion, or deletion, optionally returning the operation type.

Access

Public

Syntax

*tvcontrol.instancename.***of_CanUndo** ({ *undotype* })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
undotype (optional)	String into which the function places the type of the last operation (passed by reference): ♦ **UNDO_EDIT or "Edit"** Edit ♦ **UNDO_INSERT or "Insert"** Insertion ♦ **UNDO_DELETE or "Delete"** Deletion

Return value

Boolean. Returns TRUE if the last operation can be undone and FALSE if it cannot.

Examples

This example calls the of_CanUndo function:

```
IF tv_1.inv_levelsource.of_CanUndo() THEN
   MessageBox("TreeView", "Can undo")
ELSE
   MessageBox("TreeView", "Cannot undo")
END IF
```

of_ClearUndo

Description

Clears the TreeView's undo properties.

Access

Protected

Syntax

tvcontrol.instancename.**of_ClearUndo ( )**

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_ResetTree function:

```
...
this.of_clearundo()
...
```

of_ConfirmDelete

Description

Displays a message box allowing the user to confirm a deletion.

Access

Public

Syntax

tvcontrol.instancename.**of_ConfirmDelete** (*amount*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
amount	Long specifying the number of rows to be deleted

Return value

Integer. Returns 1 if deletion should proceed, 2 if deletions should not proceed, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the u_tvs of_DeleteItem event:

```
...
if inv_levelsource.of_IsConfirmOnDelete() then
   li_confirm = inv_levelsource.of_ConfirmDelete(1)
      if li_confirm <> CONTINUE_ACTION then
         return PREVENT_ACTION
...
```

of_ConvertToRow

Description

Converts an array of data into a row in the DataStore.

Access

Public

Syntax

tvcontrol.instancename.**of_ConvertToRow** (*parent, colvalues, datastore, row*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
parent	Long specifying the handle of the TreeView parent to which the row is added

Argument	Description
colvalues	Any array containing the data to be used for the new row
datastore	N_ds instance into which the function places the new row (passed by reference)
row	Long into which the function places the new row number (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You do not need to add a row to the associated DataStore before calling this function.

Examples

This example is from the u_tvs of_AddItem function:

```
...
If inv_levelsource.of_ConvertToRow(al_parent,  &
  aa_columnvalues, lds_datastore, ll_row) <> 1 Then
    Destroy lds_datastore
    Return -1
End If
...
```

of_GetArgs

Description

Retrieves the retrieval arguments for a specified TreeView level.

Access

Public

Syntax

tvcontrol.instancename.**of_GetArgs** (*parent, level, arguments*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
parent	Long containing the handle of the parent of the current TreeView item
level	Integer specifying the current TreeView level
arguments	Twenty-element array of the Any data type into which the function places the retrieval arguments

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_tvs pfc_Retrieve event:

```
...
If isvalid(inv_levelsource) then
   li_level = of_GetNextLevel(al_parent)
   inv_levelsource.of_getargs(al_Parent,  &
      li_Level, aa_Arg)
End If
...
```

of_GetDataObject

Description

Retrieves the DataWindow object used as the level's data source.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDataObject** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the level for which the function returns the DataWindow object

Return value

String. Returns the name of the DataWindow object used to store data for display in the TreeView.

Examples

This example calls the of_GetDataObject function:

```
String   ls_dataobject

ls_dataobject =  &
   tv_1.inv_levelsource.of_GetDataObject(1)
MessageBox("TreeView", "DataWindow object is: "  &
   + ls_dataobject)
```

of_GetDataRow

Description

Retrieves from the DataStore the row that corresponds to the specified TreeView item.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDataRow** (*handle, datasource, row*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
handle	Long specifying the handle for the TreeView item
datasource	N_ds specifying the DataStore that contains the row (passed by reference)
row	Long specifying the row in *datasource* that corresponds to *handle* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

By using this function, you can update rows in the DataSource.

Examples

This example is from the pfc_EndLabelEdit event:

```
...
If of_GetDataRow(al_handle, lds_obj, ll_Row) = -1 Then
   Return -1
End If
...
```

of_GetDataSource

Description

Retrieves a reference to the DataStore associated with a specified TreeView level.

Access

Public

Syntax

tvcontrol.instancename.**of_GetDataSource** (*level, datastore*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the level for which a DataStore reference is returned

Argument	Description
datastore	N_ds variable into which the function places a reference to the DataStore associated with *level* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if the DataStore is not found.

Examples

This example calls the of_GetDataSource function:

```
Long  ll_current, ll_return
TreeViewItem   ltvi_item
n_ds  lds_datastore

ll_current = tv_test.FindItem (CurrentTreeItem!, 0)
ll_return = tv_test.GetItem(ll_current, ltvi_item)

ll_return = tv_test.inv_levelsource.of_GetDataSource &
   (ltvi_item.level, lds_datastore)
IF ll_return = -1 THEN
   MessageBox("TreeView", "DataStore not found")
ELSE
   MessageBox("TreeView", &
      "Current level is: " &
      + String(ltvi_item.level) &
      + "~r~nDataWindow object is: " &
      + lds_datastore.DataObject)
END IF
```

of_GetItemForData

Description

Retrieves a handle to the TreeView item associated with a specified DataStore row.

Access

Public

Syntax

tvcontrol.instancename.**of_GetItemForData** (*datastore*, *row* {, *level* })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
datastore	N_ds variable specifying the DataStore containing the row whose associated TreeView item handle is returned (passed by reference)
row	Long specifying the row in *datastore* whose associated TreeView item handle is returned
level (optional)	Integer specifying the level to be searched. Specify 0 to search the entire tree (the default is 0)

Return value

Long. Returns the TreeView item handle corresponding to *row* if the function succeeds, 0 if the item was not found, and -1 if an error occurs.

Examples

This example is from the u_tvs pfc_RefreshLevel event:

```
...
For ll_row = 1 to ll_rowcount
   ll_handle = inv_levelsource.of_GetItemforData  &
      (lds_source, ll_row, ai_level)
      If ll_handle < 1 Then Continue
      If this.event pfc_RefreshItem  &
         (ll_Handle) = -1 Then Return -1
   Next
...
```

of_GetItemHandle

Description

Retrieves the handle for a specified row in the DataStore.

Access

Public

Syntax

tvcontrol.instancename.**of_GetItemHandle** (*row*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
row	Long specifying the row for which the function returns the item handle

Return value

Long. Returns the item handle if the function succeeds, 0 if the item was not found, and -1 if an error occurs.

of_GetLabelColumn

Description

Retrieves the name of the column used as the TreeView item label.

Access

Public

Syntax

tvcontrol.instancename.**of_GetLabelColumn** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the level for which the function returns the column used as the item label

Return value

String. Returns the data source column name if the function succeeds and ! if an error occurs.

Examples

This example calls the of_GetLabelColumn function:

```
String  ls_col

ls_col = tv_1.inv_levelsource.of_GetLabelColumn()
MessageBox("TreeView", "Label column is: "  &
   + ls_col)
```

of_GetLevel

Description

Retrieves the level for a specified TreeView item.

Access

Public

Syntax

tvcontrol.instancename.**of_GetLevel** (*handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
handle	Either of the following: ♦ A long specifying the TreeView item handle for which the function returns the level ♦ An n_ds instance containing the DataStore for which the function returns the associated level

Return value

Integer. Returns the level for *handle* if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetLevel function:

```
Long  ll_currenttvitem
Integer  li_level

ll_currenttvitem = tv_1.FindItem(currenttreeitem!, 0)
li_level = tv_1.inv_levelsource.of_GetLevel  &
  (ll_currenttvitem)
MessageBox("TreeView level", "Level is: "  &
  + String(li_level))
```

of_GetLevelAttributes

Description

Retrieves the attributes used to populate the specified level.

Access

Public

Syntax

tvcontrol.instancename.**of_GetLevelAttributes** (*level*, *tvattrib*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns attributes
tvattrib	N_cst_tvsrvattrib into which the function places the attributes (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_tvs pfc_PreInsert event:

```
...
If isvalid(inv_levelsource) then
   li_newlevel = of_GetNextLevel(al_parent)
   li_maxlevels = inv_levelsource.of_GetLevelCount()
   inv_levelsource.of_GetLevelAttributes  &
      (li_newlevel, lnv_attrib)
...
```

of_GetLevelCount

Description

Retrieves the number of levels defined for the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_GetLevelCount** ( )

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Return value

Integer. Returns the number of levels defined for the TreeView if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetLevelCount function:

```
Integer  li_levels

li_levels = tv_1.inv_levelsource.of_GetLevelCount()
MessageBox("TreeView", "Level count is: "  &
   + String(li_levels))
```

of_GetMethod

Description

Retrieves the method used to populate the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_GetMethod** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control

Argument	Description
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns the population method

Return value

String. Returns the method used to populate the TreeView:

DataObjectData
DataStoreControl
DataWindowControl
ImportFile
PowerObject
Retrieve
SQL

Usage

The of_Register function controls the method used to populate the TreeView.

Examples

This example calls the of_GetMethod function:

```
String  ls_method

ls_method = tv_1.inv_levelsource.of_GetMethod(1)
MessageBox("TreeView", "Method is: " + ls_method)
```

of_GetObjects

Description

Retrieves a list of updatable data sources.

Access

Public

Syntax

*tvcontrol.instancename.***of_GetObjects** (*objects*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
objects	PowerObject array into which the function places references to updatable objects (passed by reference)

Return value

Integer. Returns the number of elements in the *objects* array if the function succeeds and -1 if an error occurs.

Usage

DataStores are the only updatable data source.

Examples

This example calls the of_GetObjects function:

```
PowerObject  lpo_obj[ ]
Integer  li_return, li_count
n_ds  lds_data
String  ls_message

li_return = tv_1.inv_levelsource.of_GetObjects  &
   (lpo_obj)
FOR li_count = 1 to li_return
  IF lpo_obj[li_count].TypeOf() = DataStore! THEN
    lds_data = lpo_obj[li_count]
    ls_message += lds_data.DataObject
    ls_message += "~r~n"
   END IF
NEXT
MessageBox("TreeView", ls_message)
```

of_GetOverlayPictureColumn

Description

Retrieves the name or number of the column that specifies the overlay picture.

Access

Public

Syntax

tvcontrol.instancename.**of_GetOverlayPictureColumn** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns the overlay picture column

Return value

String. Returns the name of the column that specifies the overlay picture if the function succeeds and ! if an error occurs.

Examples

This example calls the of_GetOverlayPictureColumn function:

```
String  ls_column

ls_column =  &
```

```
    tv_1.inv_levelsource.of_GetOverlayPictureColumn(1)
MessageBox("TreeView", "Overlay is: " + ls_column)
```

of_GetPictureColumn

Description

Retrieves the name or number of the column that specifies the picture used with unselected items.

Access

Public

Syntax

tvcontrol.instancename.**of_GetPictureColumn** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns the picture column

Return value

String. Returns the name of the column that specifies the picture if the function succeeds and ! if an error occurs.

Examples

This example calls the of_GetPictureColumn function:

```
String  ls_column

ls_column = &
    tv_1.inv_levelsource.of_GetPictureColumn(1)
MessageBox("TreeView", "Picture is: " + ls_column)
```

of_GetRetrieveArgs

Description

Retrieves the retrieval definition for the specified TreeView level.

Access

Public

Syntax

tvcontrol.instancename.**of_GetRetrieveArgs** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
level	Integer specifying the TreeView level for which the function returns the retrieval arguments

Return value

String. Returns the retrieval definition if the function succeeds and ! if an error occurs.

Examples

This example calls the of_GetRetrieveArgs function:

```
String  ls_args

ls_args =  &
   tv_1.inv_levelsource.of_GetRetrieveArgs(2)
MessageBox("TreeView", "Retrieval spec: " + ls_args)
```

of_GetSelectedPictureColumn

Description

Retrieves the name or number of the column that specifies the selected picture.

Access

Public

Syntax

tvcontrol.instancename.**of_GetSelectedPictureColumn** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns the selected picture column

Return value

String. Returns the name of the column that specifies the selected picture if the function succeeds and ! if an error occurs.

Examples

This example calls the of_GetSelectedPictureColumn function:

```
String  ls_column

ls_column =  &
   tv_1.inv_levelsource.of_GetSelectedPictureColumn(1)
MessageBox("TreeView", "Selected is: " + ls_column)
```

of_GetStatePictureColumn

Description

Retrieves the name or number of the column that specifies the state picture.

Access

Public

Syntax

tvcontrol.instancename.**of_GetStatePictureColumn** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns the state picture column

Return value

String. Returns the name of the column that specifies the state picture if the function succeeds and ! if an error occurs.

Examples

This example calls the of_GetStatePictureColumn function:

```
String  ls_column

ls_column =  &
   tv_1.inv_levelsource.of_GetStatePictureColumn(2)
MessageBox("TreeView", "State picture is: " +  &
   ls_column)
```

of_GetTransObject

Description

Retrieves the Transaction object for a specified level.

Access

Public

Syntax

tvcontrol.instancename.**of_GetTransObject** (*level, trans*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level for which the function returns the population method

Argument	Description
trans	N_tr into which the function places a reference to the Transaction object defined for *level* (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if there was no valid Transaction object for *level*.

Examples

This example calls the of_GetTransObject function:

```
Integer  li_return
n_tr  ltr_trans

li_return = tv_1.inv_levelsource.of_GetTransObject  &
   (1, ltr_trans)
...
```

of_InsertItem

Description

Adds a new item to the TreeView using a row from the data source.

Access

Public

Syntax

tvcontrol.instancename.**of_InsertItem** (*parent, datastore, row, tvitem, position, after*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
parent	Long specifying the handle of the parent TreeView item
datastore	N_ds containing the data for the new item. This can be either the DataStore maintained by n_cst_tvsrv_levelsource or another DataStore. If this is another DataStore, the associated DataWindow objects must be the same and the function also adds the row to the DataStore maintained by n_cst_tvsrv_levelsource (passed by reference)
row	Long specifying the row containing the new item
tvitem	TreeViewItem containing the item to be added to the TreeView

Argument	Description
position	String specifying the position under the current parent in which to insert the new item. Values are: ♦ First ♦ Last ♦ Sort ♦ After
after	Long containing the handle of the TreeView item after which the function adds the new item. The function uses this argument if you specify After for *position*

Return value

Long. Returns the handle of the new item if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the u_tvs pfc_InsertItem event:

```
...
If isvalid(inv_levelsource) then
   Return inv_levelsource.of_insertitem  &
      (al_parent, ads_source, al_row, ltvi_new,  &
          as_position, al_handle)
End If
...
```

of_IsConfirmOnDelete

Description

Reports whether deletions are confirmed.

Access

Public

Syntax

tvcontrol.instancename.**of_IsConfirmOnDelete** ( )

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Return value

Boolean. Returns TRUE if deletions are confirmed and FALSE if they are not.

Examples

This example calls the of_IsConfirmOnDelete function:

```
IF tv_1.inv_levelsource.of_IsConfirmOnDelete() THEN
   MessageBox("TreeView", "Confirm enabled")
ELSE
   MessageBox("TreeView", "Confirm disabled")
END IF
```

of_IsRecursiveLevel

Description — Reports whether the specified level is recursive.

Access — Public

Syntax — *tvcontrol.instancename*.**of_IsRecursiveLevel** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level the function tests for recursive processing

Return value — Boolean. Returns TRUE if *level* is recursive and FALSE if it is not.

Examples — This example calls the of_IsRecursiveLevel function:

```
IF tv_1.inv_levelsource.of_IsRecursiveLevel(2) THEN
   MessageBox("TreeView", "Is recursive")
ELSE
   MessageBox("TreeView", "Is not recursive")
END IF
```

of_ParseArgs

Description — Creates an array of retrieval arguments from a passed string.

Access — Protected

Syntax — *tvcontrol.instancename*.**of_ParseArgs** (*argument, level, arglevels, argcolumns*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control

Argument	Description
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
argument	String containing the retrieval argument
level	Integer specifying the TreeView level for which the arguments will be used
arglevels	Integer array into which the function places the TreeView levels used to create the retrieval argument (passed by reference)
argcolumns	String array into which the function places the columns used in the retrieval argument (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_GetArgs function:

```
...
of_ParseArgs(inv_attrib[li_Index].is_RetrieveArgs, &
   ai_Level, li_RetrieveArgLevel,  &
      ls_RetrieveArgColumn)
...
```

of_RefreshItem

Description

Refreshes a TreeView item. This includes discarding all children and resetting label and data attributes from the DataStore.

Access

Public

Syntax

tvcontrol.instancename.**of_RefreshItem** (*handle* {, *row* })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
handle	Long containing the handle of the item to be refreshed

Argument	Description
row (optional)	Long optionally specifying the DataStore row associated with the item to be refreshed. The default is the row for the current item. If you have changed any key columns in the DataStore, you must specify this argument

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the of_RefreshLevel function:

```
...
Do
   If of_RefreshItem(ll_Handle) = -1 Then Return -1
...
```

of_RefreshLevel

Description

Refreshes all items at a specified TreeView level. This includes discarding all children and resetting label and data attributes from the DataStore.

Access

Public

Syntax

tvcontrol.instancename.**of_RefreshLevel** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the level to be refreshed

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_RefreshLevel function:

```
Integer  li_return
Long     ll_current
TreeViewItem  ltvi_item

ll_current = tv_test.FindItem (CurrentTreeItem!, 0)
li_return = tv_test.GetItem(ll_current, ltvi_item)
li_return = tv_test.inv_levelsource.of_RefreshLevel &
   (ltvi_item.level)
IF li_return = -1 THEN
```

```
    MessageBox("TreeView", "Refresh failed")
ELSE
    gnv_app.of_GetFrame().SetMicroHelp &
        ("Refresh succeeded for level " &
            + String(ltvi_item.level))
END IF
```

of_Register

Registers a data source for the TreeView control. There are six syntaxes.

To	Use
Display data retrieved via a DataWindow object	Syntax 1
Display data retrieved via a SQL statement	Syntax 2
Display data from a DataWindow control	Syntax 3
Display data from a DataStore	Syntax 4
Display passed data	Syntax 5
Display data from a file	Syntax 6

Another version of of_Register
There is also an undocumented version of of_Register that is for internal use.

Syntax 1

To display data retrieved via a DataWindow object

Description

Establishes a TreeView level's data source by retrieving rows using the specified DataWindow object.

Access

Public

Syntax

tvcontrol.instancename.**of_Register** (*level*, *labelcolumn*, *retrievespec*, *dwobject* {, *trans*, *filterspec* })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
level	Integer specifying the TreeView level associated with this data source
labelcolumn	String specifying the *dwobject* column that u_tvs uses for TreeView item labels
retrievespec	String containing the retrieval specification for *dwobject*
dwobject	String specifying the DataWindow object used to retrieve data
trans (optional)	N_tr Transaction object used to retrieve rows for *dwobject* If you do not specify this argument, *dwobject* must have been saved with the data
filterspec (optional)	String containing the filter specification for *dwobject*

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** *Dwobject* had no key columns assigned
- **-3** A previous level was marked as recursive

Usage

Call this function once for each TreeView level.

The *retrieveargs* argument must be in the format:

:scope.level.column

where:

- *Scope* specifies either level or parent:
  - **Level** An absolute level number
  - **Parent** A number relative to the current level
- *Level* specifies an absolute or relative level number, depending on what you specify for scope:

Scope specification	Level specification
Level	The value you specify indicates an absolute level number. For example, **:level.1.emp_name** indicates that the retrieval argument comes from the emp_name column of the item's level-1 ancestor

Scope specification	Level specification
Parent	The value you specify indicates a level relative to the current level. For example, **:parent.2.emp_name** indicates that the retrieval argument comes from the emp_name column of the ancestor two levels above

- *Column* specifies the DataWindow object column name from which to obtain the values used in retrieval arguments

Examples

This example calls the of_Register function:

```
this.of_SetLevelSource(TRUE)
this.inv_levelsource.of_Register(1, "dept_name",  &
  "", "d_deptlist", SQLCA, "")
this.inv_levelsource.of_SetPictureColumn(1, "1")
this.inv_levelsource.of_SetSelectedPictureColumn  &
   (1, "2")
this.inv_levelsource.of_Register(2, "emp_lname",  &
  ":parent.1.dept_id", "d_empbydept", SQLCA, "")
this.inv_levelsource.of_SetPictureColumn(2, "4")
this.inv_levelsource.of_SetSelectedPictureColumn  &
   (2, "5")

this.event pfc_populate(0)
```

Syntax 2 **To display data retrieved via a SQL statement**

Description

Establishes a TreeView level's data source by retrieving rows from the database using the specified SQL statement.

Access

Public

Syntax

tvcontrol.instancename.**of_Register** (*level*, *labelcolumn*, *retreivespec*, *trans*, *sql*, *filterspec*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level associated with this data source

Argument	Description
labelcolumn	String specifying the column that u_tvs uses for TreeView item labels
retrievespec	String containing the retrieval arguments if required
trans	N_tr Transaction object used to retrieve rows for *sql*
sql	String specifying the SQL SELECT statement used to create the TreeView level's data source
filterspec (optional)	String containing the filter specification if required

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** A specified key column did not exist
- **-3** A previous level was marked as recursive

Usage

Call this function before calling other functions that customize TreeView display.

Examples

This example calls the of_Register function:

```
String  ls_sql

this.of_SetLevelSource(TRUE)
ls_sql = "Select * from department ORDER BY dept_id;"
this.inv_levelsource.of_Register(1, "dept_name",  &
   "", SQLCA, ls_sql, "")
this.inv_levelsource.of_SetPictureColumn(1, "1")
this.inv_levelsource.of_SetSelectedPictureColumn  &
   (1, "2")
this.inv_levelsource.of_Register(2, "emp_lname",  &
  ":parent.1.dept_id", "d_empbydept", SQLCA, "")
this.inv_levelsource.of_SetPictureColumn(2, "4")
this.inv_levelsource.of_SetSelectedPictureColumn  &
  (2, "5")

this.event pfc_populate(0)
```

Syntax 3

To display data from a DataWindow control

Description

Establishes a TreeView level's data source using rows contained in the passed DataWindow control.

Access

Public

Syntax

*tvcontrol.instancename.***of_Register** (*level, labelcolumn, retrievespec, dwcontrol, filterspec*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level associated with this data source
labelcolumn	String specifying the column that u_tvs uses for TreeView item labels
retrievespec	String containing the retrieval arguments if required
dwcontrol	DataWindow control containing data to use in the TreeView level
filterspec (optional)	String containing the filter specification if required

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** The DataWindow object associated with *dwcontrol* had no key columns assigned
- **-3** A previous level was marked as recursive

Usage

Call this function before calling other functions that customize TreeView display.

Examples

This example calls the of_Register function:

```
this.of_SetLevelSource(TRUE)
this.inv_levelsource.of_Register(1, "dept_name",  &
   "", dw_1, "")
this.inv_levelsource.of_SetPictureColumn(1, "1")
this.inv_levelsource.of_SetSelectedPictureColumn  &
```

```
    (1, "2")
this.inv_levelsource.of_Register(2, "emp_lname",  &
    ":parent.1.dept_id", "d_empbydept", SQLCA, "")
this.inv_levelsource.of_SetPictureColumn(2, "4")
this.inv_levelsource.of_SetSelectedPictureColumn  &
    (2, "5")

this.event pfc_populate(0)
```

Syntax 4 **To display data from a DataStore**

Description

Establishes a TreeView level's data source using rows in the passed DataStore.

Access

Public

Syntax

tvcontrol.instancename.**of_Register** (*level*, *labelcolumn*, *retrievespec*, *datastore*, *filterspec*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level associated with this data source
labelcolumn	String specifying the column that u_tvs uses for TreeView item labels
retrievespec	String containing the retrieval arguments if required
datastore	DataStore containing data to use in the TreeView level
filterspec (optional)	String containing the filter specification if required

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** The DataWindow object associated with *datastore* had no key columns assigned
- **-3** A previous level was marked as recursive

Usage

Call this function before calling other functions that customize TreeView display.

Examples

This example calls the of_Register function:

```
tv_1.of_SetLevelSource(TRUE)
tv_1.inv_levelsource.of_Register(1, "dept_name",  &
  "", ids_datastore, "")
tv_1.inv_levelsource.of_SetPictureColumn(1, "1")
tv_1.inv_levelsource.of_SetSelectedPictureColumn  &
   (1, "2")
tv_1.inv_levelsource.of_Register(2, "emp_lname",  &
  ":parent.1.dept_id", "d_empbydept", SQLCA, "")
tv_1.inv_levelsource.of_SetPictureColumn(2, "4")
tv_1.inv_levelsource.of_SetSelectedPictureColumn  &
  (2, "5")

tv_1.event pfc_populate(0)
```

Syntax 5

To display passed data

Description

Establishes a TreeView level's data source using passed data.

Access

Public

Syntax

tvcontrol.instancename.**of_Register** (*level*, *labelcolumn*, *retrievespec*, *dwobject*, *data*, *filterspec*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level associated with this data source
labelcolumn	String specifying the column that u_tvs uses for TreeView item labels
retrievespec	String containing the retrieval arguments if required
dwobject	String specifying the DataWindow object whose columns correspond to *data*
data	PowerObject array containing the data to display in the TreeView level

Argument	Description
filterspec (optional)	String containing the filter specification if required

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** *Dwobject* had no key columns assigned
- **-3** A previous level was marked as recursive

Usage

Call this function before calling other functions that customize TreeView display.

Elements must match
Errors will occur if the elements of the *data* structure do not exactly match the columns of *dwobject*.

Examples

This example calls the of_Register function:

```
tv_1.of_SetLevelSource(TRUE)
tv_1.inv_levelsource.of_Register(1, "dept_name",  &
  "", "d_deptlist", istr_dept, "")
tv_1.inv_levelsource.of_SetPictureColumn(1, "1")
tv_1.inv_levelsource.of_SetSelectedPictureColumn  &
  (1, "2")
tv_1.inv_levelsource.of_Register(2, "emp_lname",  &
  ":parent.1.dept_id", "d_empbydept", SQLCA, "")
tv_1.inv_levelsource.of_SetPictureColumn(2, "4")
tv_1.inv_levelsource.of_SetSelectedPictureColumn  &
  (2, "5")

tv_1.event pfc_populate(0)
```

Syntax 6 **To display data from a file**

Description

Establishes a TreeView level's data source using data from a file.

Access

Public

Syntax

tvcontrol.instancename.**of_Register** (*level*, *labelcolumn*, *retrievespec*, *dwobject*, *file*, *filterspec*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level associated with this data source
labelcolumn	String specifying the column that u_tvs uses for TreeView item labels
retrievespec	String containing the retrieval arguments if required
dwobject	String specifying the DataWindow object whose columns correspond to the data in *file*
file	String specifying the file to be used as the TreeView level's data source
filterspec (optional)	String containing the filter specification if required

Return value

Integer. Returns values as follows:

- **1** Success
- **-1** Argument validation error
- **-2** *Dwobject* had no key columns assigned
- **-3** A previous level was marked as recursive

Usage

Call this function before calling other functions that customize TreeView display.

Elements must match
Errors will occur if the elements of the lines in *filename* do not exactly match the columns of *dwobject*.

Examples

This example calls the of_Register function:

```
tv_1.of_SetLevelSource(TRUE)
ls_file = "c:\pb6i32\pfc\deptlist.txt"
tv_1.inv_levelsource.of_Register(1, "dept_name",  &
  "", "d_deptlist", ls_file, "")
tv_1.inv_levelsource.of_SetPictureColumn(1, "1")
tv_1.inv_levelsource.of_SetSelectedPictureColumn  &
```

```
    (1, "2")
tv_1.inv_levelsource.of_Register(2, "emp_lname",  &
    ":parent.1.dept_id", "d_empbydept", SQLCA, "")
tv_1.inv_levelsource.of_SetPictureColumn(2, "4")
tv_1.inv_levelsource.of_SetSelectedPictureColumn  &
    (2, "5")

tv_1.event pfc_populate(0)
```

of_RegisterDataSource

Description

Registers the TreeView data source with n_cst_dwcache (the PFC caching service).

Access

Protected

Syntax

tvcontrol.instancename.**of_RegisterDataSource** (*level, method, dwobject, trans, sql, data, dwcontrol, datastore, file*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level associated with this data source
method	String specifying the data source population method. Values are: Retrieve DataObjectData PowerObject DataWindowControl DataStoreControl ImportFile SQL
dwobject	String specifying the DataWindow object used by the data source
trans	N_tr specifying the Transaction object used by the data source
sql	String specifying the SQL statement used as the data source
data	PowerObject array containing data used as the data source

Argument	Description
dwcontrol	DataWindow control whose associated DataWindow object and data is used as the data source
datastore	DataStore whose associated DataWindow object and data is used as the data source
file	String specifying the file used as the data source

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Register function:

```
...
li_rc = of_registerdatasource(ai_level,  &
   as_method, as_DWobjectname, atr_obj, &
      as_sql, apo_data, adw_control, ads_control, &
         as_importfile)
...
```

of_RemoveChildren

Description

Removes all TreeView items beneath the specified item. This function also removes rows from the associated data source.

Access

Public

Syntax

*tvcontrol.instancename.***of_RemoveChildren** (*handle*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
handle	Long specifying the TreeView item whose children are removed

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the of_DeleteItem function:

```
...
If of_RemoveChildren(al_Handle) = -1 Then Return -1
```

```
...
```

of_Reset

Description: Clears all data from the data source.

Access: Public

Syntax: *tvcontrol.instancename*.**of_Reset** ({ *level* })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level (optional)	Integer specifying the TreeView level to be cleared. Specify 0 to reset all levels (the default is 0)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: If *level* is a recursive level, the function resets all levels using that data source.

Examples: This example is from the of_ResetTree function:

```
...
If this.of_reset() <> 1 Then Return -1
...
```

of_ResetTree

Description: Clears data from the TreeView and the associated data source.

Access: Public

Syntax: *tvcontrol.instancename*.**of_ResetTree** ( )

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples: This example calls the of_ResetTree function:

```
tv_1.inv_levelsource.of_ResetTree()
```

of_ResetUpdate

Description

Resets the update flags in the DataStore that contains TreeView data.

Access

Public

Syntax

tvcontrol.instancename.**of_ResetUpdate** ({ *level* })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose data source's update flags are to be reset. Specify 0 to reset flags for all levels (the default is 0)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

If *level* is a recursive level, the function resets update flags for all levels using that data source.

Examples

This example calls the of_ResetUpdate function:

```
tv_1.inv_levelsource.of_ResetUpdate()
```

of_Retrieve

Description

Retrieves TreeView data from rows in the DataStore.

Access

Public

Syntax

tvcontrol.instancename.**of_Retrieve** (*level, arguments, data*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose data source is retrieved
arguments	Twenty-element Any array containing retrieval arguments

Argument	Description
data	N_ds into which the function retrieves data (passed by reference)

Return value

Integer. Returns the number of items retrieved if the function succeeds and -1 if an error occurs.

Usage

Specify retrieval arguments by extending the u_tvs pfc_Retrieve event.

Examples

This example is from the u_tvs of_Retrieve event:

```
...
If isvalid(inv_levelsource) then
   ll_rows = inv_levelsource.of_Retrieve  &
      (li_level, la_args, ads_data)
End If
...
```

of_SetCache

Description

Enables or disables n_cst_dwcache (the PFC caching service).

Access

Protected

Syntax

tvcontrol.instancename.**of_SetCache** (*boolean*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
boolean	Boolean indicating whether to enable (TRUE) or disable (FALSE) n_cst_dwcache

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the Constructor event:

```
of_SetCache(True)
...
```

of_SetConfirmOnDelete

Description

Specifies whether n_cst_tvsrv_levelsource displays a confirmation dialog box when users delete items.

Access

Public

Syntax

tvcontrol.instancename.**of_SetConfirmOnDelete** (*boolean*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
boolean	Boolean specifying whether to display (TRUE) or not display (FALSE) a deletion confirmation dialog box

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetConfirmOnDelete function:

```
...
this.inv_levelsource.of_SetConfirmOnDelete(TRUE)
this.event pfc_populateitem(0)
```

of_SetDeleteStyle

Description

Specifies whether deleted child rows are deleted or discarded.

Access

Public

Syntax

tvcontrol.instancename.**of_SetDeleteStyle** (*level, style*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose deletion style is specified
style	Integer or n_cst_tvsrv_levelsource constant specifying how to handle deleted children: ♦ **DISCARD_ROWS or 0** Discard child rows (default) ♦ **DELETE_ROWS or 1** Delete child rows

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetDeleteStyle function:

```
...
this.inv_levelsource.of_SetDeleteStyle  &
   (this.inv_levelsource.DELETE_ROWS)
this.event pfc_populateitem(0)
```

of_SetItemAttributes

Description

Sets TreeView item default properties using data from the data source.

Access

Public

Syntax

tvcontrol.instancename.**of_SetItemAttributes** (*datastore*, *row*, *item*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
datastore	N_ds containing the DataStore used to create TreeView item properties
row	Long specifying the row in *datastore* used to create TreeView item properties
item	TreeViewItem whose properties are to be set (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the u_tvs pfc_SetItemAttributes event:

```
If isvalid(inv_levelsource) then
   inv_levelsource.of_SetItemAttributes  &
      (ads_obj, al_row, atvi_item)
End If
...
```

of_SetOverlayPictureColumn

Description

Specifies the name or number of the DataWindow object column that references the overlay picture.

Access

Public

Syntax

tvcontrol.instancename.**of_SetOverlayPictureColumn** (*level, column*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose overlay picture is specified
column	String specifying one of the following: ♦ The DataWindow object column that u_tvs uses for overlay pictures ♦ A number, which causes u_tvs to use the TreeView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetOverlayPictureColumn function:

```
tv_1.inv_levelsource.of_SetOverlayPictureColumn &
   (1, "4")
```

of_SetPictureColumn

Description

Specifies the default picture.

Access

Public

Syntax

tvcontrol.instancename.**of_SetPictureColumn** (*level, column*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose picture is specified

Argument	Description
column	String specifying one of the following: ♦ The DataWindow object column that u_tvs uses for pictures. This column can contain either a filename or a number, which causes u_tvs to use the TreeView's picture index ♦ A number, which causes u_tvs to use the TreeView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetPictureColumn function:

```
...
this.inv_levelsource.of_Register(2, "emp_lname",  &
  ls_keys, "d_empbydept", SQLCA, ":parent.1.dept_id")
this.inv_levelsource.of_SetPictureColumn(2, "4")
...
```

of_SetRecursive

Description

Specifies whether a specified level is recursive.

Access

Public

Syntax

tvcontrol.instancename.**of_SetRecursive** (*level, boolean*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level
boolean	Boolean specifying whether *level* is recursive (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function after calling of_Register. A recursive level is always the lowest level specified.

A recursive level works only if a table has a recursive relationship. In the employee table, for example, there might be a recursive relationship between managers and employees, with each employee row containing a column that points to its manager's employee ID.

Examples

This example calls the of_SetRecursive function:

```
...
this.inv_levelsource.of_SetRecursive(3, TRUE)
...
```

of_SetSelectedPictureColumn

Description

Specifies the name of the DataWindow object column that references the selected picture.

Access

Public

Syntax

tvcontrol.instancename.**of_SetSelectedPictureColumn** (*level, column*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose selected picture is specified
column	String specifying one of the following: ♦ The DataWindow object column that u_tvs uses for selected pictures ♦ A number, which causes u_tvs to use the TreeView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetSelectedPictureColumn function:

```
...
this.inv_levelsource.of_Register(2, "emp_lname",  &
  ":parent.1.dept_id", "d_empbydept", SQLCA, "")
this.inv_levelsource.of_SetPictureColumn(2, "4")
this.inv_levelsource.of_SetSelectedPictureColumn  &
```

```
    (2, "5")
...
```

of_SetStatePictureColumn

Description

Specifies the name of the DataWindow object column that references the state picture.

Access

Public

Syntax

tvcontrol.instancename.**of_SetStatePictureColumn** (*level, column*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level whose state picture is specified
column	String specifying one of the following: ♦ The DataWindow object column that u_tvs uses for state pictures. This column can contain either a filename or a number, which causes u_tvs to use the TreeView's picture index ♦ A number, which causes u_tvs to use the TreeView's picture index

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call of_Register before calling this function.

Examples

This example calls the of_SetStatePictureColumn function:

```
...
this.inv_levelsource.of_Register(2, "emp_lname",  &
  ls_keys, "d_empbydept", SQLCA, ":parent.1.dept_id")
this.inv_levelsource.of_SetPictureColumn(2, "4")
this.inv_levelsource.of_SetSelectedPictureColumn  &
   (2, "5")
this.inv_levelsource.of_SetStatePictureColumn(2, "11")
...
```

of_UnRegister

Description

Removes a data source for the specified level.

Access

Public

Syntax

tvcontrol.instancename.**of_UnRegister** (*level*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
level	Integer specifying the TreeView level to unregister

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_UnRegister function:

```
...
tv_1.inv_levelsource.of_UnRegister()
...
```

of_Update

Updates data associated with a TreeView control. There are two syntaxes:

To	Use
Update a specified TreeView level	Syntax 1
Update all TreeView levels	Syntax 2

Syntax 1

Update a specified TreeView level

Description

Saves all rows in the DataStore associated with the TreeView level

Access

Public

Syntax

tvcontrol.instancename.**of_Update** (*level, accept, resetflag*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
level	Integer specifying the TreeView level whose DataStore is to be updated
accept	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database
resetflag	Boolean indicating whether the Update function resets the update flags

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Update function:

```
tv_1.inv_levelsource.of_Update(2, TRUE, TRUE)
```

Syntax 2

Update all TreeView levels

Description

Saves all rows in all DataStores associated with the TreeView.

Access

Public

Syntax

tvcontrol.instancename.**of_Update** (*accept, resetflag*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
accept	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database
resetflag	Boolean indicating whether the Update function resets the update flags

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_Update function:

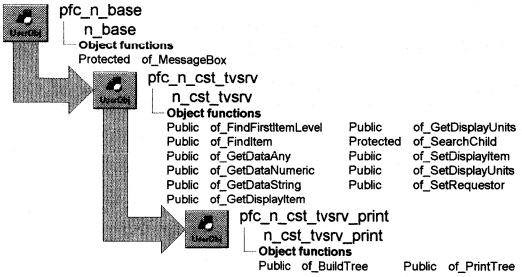
```
tv_1.inv_levelsource.of_Update(TRUE, TRUE)
```

n_cst_tvsrv_print

Description

TreeView print service.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

To use this service:

1 Enable the service by calling the u_tvs of_SetPrint function:

```
tv_1.of_SetPrint(TRUE)
```

2 Call the of_PrintTree function to print TreeView data:

```
String  ls_levelpicts[10]
String  ls_levelselpicts[10]

ls_levelpicts[1] = "dept.bmp"
ls_levelpicts[2] = "emp.bmp"
ls_levelselpicts[1] = "deptsel.bmp"
ls_levelselpicts[2] = "empsel.bmp"
tv_1.inv_print.of_PrintTree(0, TRUE, &
   ls_levelpicts, ls_levelselpicts)
```

See also

n_cst_tvsrv_levelsource

Functions

N_cst_tvsrv_print includes precoded functions:

of_BuildTree
of_PrintTree

of_BuildTree

Description

Builds the data to be printed.

Access

Public

Syntax

tvcontrol.instancename.**of_BuildTree** (*handle, pictures, levelpicts*)

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_print (the u_tvs default is inv_print)
handle	Long containing the handle of the TreeView item from which to start building the data to print
pictures	String specifying the filenames for all pictures and selected pictures
levelpicts	String specifying the filenames for pictures and selected pictures to display when using different pictures for each level

Return value

String. Returns a string used to populate a temporary DataStore, which is then printed.

Usage

Internal.

Examples

This example is from the of_PrintTree function:

```
...
lds_Print = Create n_ds
lds_Print.DataObject = "d_printtree"
If lds_Print.ImportString(of_buildtree(al_Handle,  &
   ls_Picts, ls_LevelPicts)) <= 0 Then Return -1
li_RC = lds_Print.Print(ab_CancelDialog)
...
```

of_PrintTree

Prints data in a TreeView control. There are two syntaxes:

To	Use
Print a TreeView using the same picture and selected picture for all levels	Syntax 1

To	Use
Print a TreeView specifying pictures and selected pictures for each level	Syntax 2

Another version of of_PrintTree
There is also an undocumented version of of_Register that is for internal use.

Syntax 1

Print a TreeView using the same pictures for each level

Description

Prints TreeView data using the same picture and selected picture for each level.

Access

Public

Syntax

*tvcontrol.instancename.***of_PrintTree** ({ *handle* {, *canceldialog* {, *pictname*, *selectedpictname* } } })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_print (the u_tvs default is inv_print)
handle (optional)	Long specifying the handle of the TreeView item from which to begin printing. The default is 0, which starts at the top of the TreeView hierarchy
canceldialog (optional)	Boolean specifying whether to display a Cancel dialog. The default is FALSE
pictname (optional)	String specifying the filename to display as the picture. The default is folder.bmp
selectedpictname (optional)	String specifying the filename to display as the selected picture. The default is foldopen.bmp

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The pictures specified (including the default pictures, folder.bmp and foldopen.bmp) must be in a directory that is accessible at execution time.

Examples

This example calls the of_PrintTree function:

```
tv_1.inv_print.of_PrintTree  &
   (0, TRUE, "dept.bmp", "emp.bmp")
```

Syntax 2 **Print a TreeView specifying pictures for each level**

Description

Prints TreeView data using the specified pictures and selected pictures for each level.

Access

Public

Syntax

tvcontrol.instancename.**of_PrintTree** ({ *handle* {, *canceldialog* {, *levelpicts*, *levelselectedpicts* } } })

Argument	Description
tvcontrol	Instance name of the u_tvs-based TreeView control
instancename	Instance name of n_cst_tvsrv_print (the u_tvs default is inv_print)
handle (optional)	Long specifying the handle of the TreeView item from which to begin printing. The default is 0, which starts at the top of the TreeView hierarchy
canceldialog (optional)	Boolean specifying whether to display a Cancel dialog. The default is FALSE
levelpicts (optional)	Ten-element string array specifying pictures for items in up to ten levels
levelselectedpicts (optional)	Ten-element string array specifying selected pictures for items in up to ten levels

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The pictures specified (including the default pictures, folder.bmp and foldopen.bmp) must be in a directory that is accessible at execution time.

Examples

This example calls the of_PrintTree function:

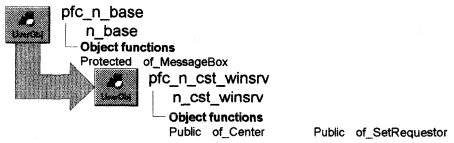
```
String  ls_levelpicts[10]
String  ls_levelselpicts[10]

ls_levelpicts[1] = "dept.bmp"
ls_levelpicts[2] = "emp.bmp"
ls_levelselpicts[1] = "deptsel.bmp"
ls_levelselpicts[2] = "empsel.bmp"
tv_1.inv_print.of_PrintTree(0, TRUE, &
   ls_levelpicts, ls_levelselpicts)
```

n_cst_winsrv

Description

N_cst_winsrv is the base window services object and serves as the ancestor object for all Window services. In addition to being the ancestor for all other Window services, w_master instantiates this user object as inv_base.

Ancestry



Library

PFCWNSRV.PBL
PFEWNSRV.PBL

Usage

Use the functions in this user object to obtain basic information about a DataWindow object and its contents.

To use this object:

1 Enable the service using the w_master of_SetBase function:

```
this.of_SetBase(TRUE)
```

2 Call base Window service functions as needed.

Descendants

n_cst_winsrv_preference
n_cst_winsrv_sheetmanager
n_cst_winsrv_statusbar

See also

n_cst_resize
w_master

Instance variables

N_cst_winsrv includes one instance variable:

Instance variable	Description	Data type	Access	Usage
iw_requestor	Window that owns the n_cst_winsrv instance	w_master	Protected	Use to access and control the associated window

Functions

N_cst_winsrv includes precoded object functions:

of_Center
of_SetRequestor

of_Center

Description

Centers the window relative to the dimensions of the current display resolution.

Access

Public

Syntax

instancename.**of_Center** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv (the w_master default is inv_base)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example from a window Open event calls the of_Center function.

```
Integer  li_return

this.of_SetBase(TRUE)
li_return = this.inv_base.of_Center()
...
```

of_SetRequestor

Description

Associates a window with n_cst_winsrv.

Access

Public

Syntax

instancename.**of_SetRequestor** (*window*)

Argument	Description
instancename	Instance name of n_cst_winsrv (the w_master default is inv_base)
window	W_master window variable containing the window to be associated with n_cst_winsrv

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the w_master of_SetBase function:

```
IF ab_Switch THEN
   IF Not IsValid(inv_base) THEN
      inv_base = Create n_cst_winsrv
      inv_base.of_SetRequestor(this)
   END IF
ELSE
   IF IsValid(inv_base) THEN
      Destroy inv_base
   END IF
END IF
```

n_cst_winsrv_preference

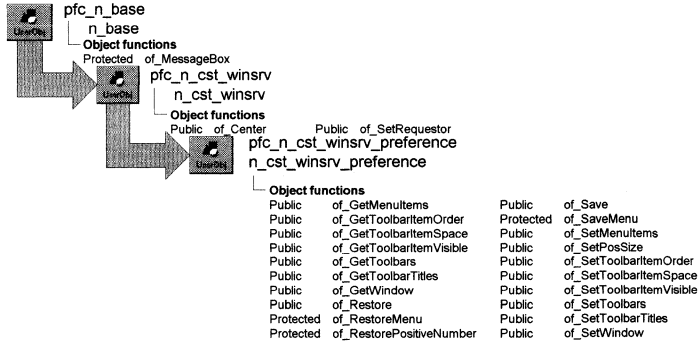
Description

Window preference service. Using this service, you can save and restore window state information using either an INI file or the Windows registry. W_master instantiates this user object as inv_preference. This service saves the following properties:

ToolBarVisible
ToolBarAlignment
ToolBarItemOrder
Toolbar size (ToolbarHeight and ToolbarWidth)
Toolbar position (ToolbarX and ToolbarY)
Window size (height and width)
Window state
Window position (X and Y)
Menu item attributes

The service provides functions allowing you to control which properties are restored.

Ancestry



Library

PFCWNSRV.PBL
PFEWNSRV.PBL

Object relationships

n_cst_conversion
n_cst_inifile
n_cst_string
n_cst_menu

Usage

To use the preference service:

1 Create an instance of n_cst_winsrv_preference by calling the w_master of_SetPreference function.

2 Specify the items to be restored by calling one or more of the following functions:

- The of_SetMenuItems function controls whether the service restores menu item properties
- The of_SetWindow function controls whether the service restores window properties
- The of_SetToolbars function controls whether the service restores toolbars
- The of_SetToolbarTitles function controls whether the service restores toolbar titles
- The of_SetToolbarItemOrder function controls whether the service restores the order of toolbar items
- The of_SetToolbarItemSpace function controls whether the service restores the ToolbarItemOrder property
- The of_SetToolbarItemVisible function controls whether the service restores the ToolbarItemVisible property

3 Call the of_Restore function to reapply previous settings.

4 Before closing the window, call the of_Save function to save window settings in the registry or an INI file.

See also

n_cst_winsrv
n_cst_resize
n_cst_winsrv_sheetmanager
n_cst_winsrv_statusbar
w_master

Instance variables

N_cst_winsrv_preference includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_menuitems	Controls whether the service restores menu item properties	Boolean	Protected	Access through of_GetMenuItems and of_SetMenuItems
ib_toolbaritemorder	Controls whether the service restores toolbar item order	Boolean	Protected	Access through of_GetToolbarItemOrder and of_SetToolbarItemOrder

Instance variable	Description	Data type	Access	Usage
ib_toolbaritemspace	Controls whether the service restores the ToolbarItemSpace property	Boolean	Protected	Access through of_GetToolbarItemSpace and of_SetToolbarItemSpace
ib_toolbaritemvisible	Controls whether the service restores the ToolbarItemVisible property	Boolean	Protected	Access through of_GetToolbarItemVisible and of_SetToolbarItemVisible
ib_toolbars	Controls whether the service restores toolbars	Boolean	Protected	Access through of_GetToolbars and of_SetToolbars
ib_toolbartitles	Controls whether the service restores toolbar titles	Boolean	Protected	Access through of_GetToolbarTitles and of_SetToolbarTitles
ib_window	Controls whether the service restores window properties	Boolean	Protected	Access through of_GetWindow and of_SetWindow
ii_normalstate_height	Stores the window's height in normal display	Integer	Protected	Internal
ii_normalstate_width	Stores the window's width in normal display	Integer	Protected	Internal
ii_normalstate_x	Stores the window's *x* position in normal display	Integer	Protected	Internal
ii_normalstate_y	Stores the window's *y* position in normal display	Integer	Protected	Internal

Functions

N_cst_winsrv_preference includes precoded object functions:

of_GetMenuItems	of_Save
of_GetToolbarItemOrder	of_SaveMenu
of_GetToolbarItemSpace	of_SetMenuItems
of_GetToolbarItemVisible	of_SetPosSize

of_GetToolbars	of_SetToolbarItemOrder
of_GetToolbarTitles	of_SetToolbarItemSpace
of_GetWindow	of_SetToolbarItemVisible
of_Restore	of_SetToolbars
of_RestoreMenu	of_SetToolbarTitles
of_RestorePositiveNumber	of_SetWindow

of_GetMenuItems

Description

Reports whether the service restores menu items.

Access

Public

Syntax

instancename.**of_GetMenuItems** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if menu items are restored and FALSE if they are not.

Examples

This example calls the of_GetMenuItems function:

```
IF parent.inv_preference.of_GetMenuItems() THEN
   MessageBox("Preferences", &
      "Menu item restore is enabled")
ELSE
   MessageBox("Preferences", &
      "Menu item restore is disabled")
END IF
```

of_GetToolbarItemOrder

Description

Reports whether the service restores toolbar item order.

Access

Public

Syntax

instancename.**of_GetToolbarItemOrder** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if toolbar item order is restored and FALSE if it is not.

Examples

This example calls the of_GetToolbarItemOrder function:

```
IF parent.inv_preference.of_GetToolbarItemOrder() &
   THEN
      MessageBox("Preferences", &
         "Toolbar item order restore is enabled")
ELSE
   MessageBox("Preferences", &
      "Toolbar item order restore is disabled")
END IF
```

of_GetToolbarItemSpace

Description

Reports whether the service restores the ToolbarItemSpace property.

Access

Public

Syntax

instancename.**of_GetToolbarItemSpace** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if the ToolbarItemSpace property is restored and FALSE if it is not.

Examples

This example calls the of_GetToolbarItemSpace function:

```
IF parent.inv_preference.of_GetToolbarItemSpace() &
   THEN
      MessageBox("Preferences", &
         "ToolbarItemSpace restore is enabled")
ELSE
   MessageBox("Preferences", &
      "ToolbarItemSpace restore is disabled")
END IF
```

of_GetToolbarItemVisible

Description

Reports whether the service restores the ToolbarItemVisible property.

Access

Public

Syntax

instancename.**of_GetToolbarItemVisible** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if the ToolbarItemVisible property is restored and FALSE if it is not.

Examples

This example calls the of_GetToolbarItemVisible function:

```
IF parent.inv_preference.of_GetToolbarItemVisible() &
   THEN
      MessageBox("Preferences", &
         "ToolbarItemVisible restore is enabled")
ELSE
   MessageBox("Preferences", &
      "ToolbarItemVisible restore is disabled")
END IF
```

of_GetToolbars

Description

Reports whether the service restores toolbars.

Access

Public

Syntax

instancename.**of_GetToolbars ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if toolbars are restored and FALSE if they are not.

Examples

This example calls the of_GetToolbars function:

```
IF parent.inv_preference.of_GetToolbars() THEN
   MessageBox("Preferences", &
      "Toolbar restore is enabled")
ELSE
   MessageBox("Preferences", &
      "Toolbar restore is disabled")
END IF
```

of_GetToolbarTitles

Description

Reports whether the service restores toolbar titles.

Access

Public

Syntax

instancename.**of_GetToolbarTitles** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if toolbar titles are restored and FALSE if they are not.

Examples

This example calls the of_GetToolbarTitles function:

```
IF parent.inv_preference.of_GetToolbarTitles() THEN
   MessageBox("Preferences", &
      "Toolbar titles restore is enabled")
ELSE
   MessageBox("Preferences", &
      "Toolbar titles restore is disabled")
END IF
```

of_GetWindow

Description

Reports whether the service restores window properties.

Access

Public

Syntax

instancename.**of_GetWindow** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Boolean. Returns TRUE if window properties are restored and FALSE if they are not.

Examples

This example calls the of_GetWindow function:

```
IF parent.inv_preference.of_GetWindow() THEN
   MessageBox("Preferences", &
      "Window property restore is enabled")
ELSE
```

```
        MessageBox("Preferences", &
            "Window property restore is disabled")
    END IF
```

of_Restore

Restores window, toolbar, and menu settings from the registry or an INI file. There are two syntaxes:

To	Use
Restore settings from the registry	Syntax 1
Restore settings from an INI file	Syntax 2

Other versions of of_Resize
There are also two undocumented versions of of_Resize, which are for internal use.

Syntax 1

To restore settings from the registry

Description

Restores window size, position, toolbar settings, and menu item settings from the registry.

Access

Public

Syntax

instancename.**of_Restore** (*registrykey*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
registrykey	String specifying the registry key from which to restore settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function in a window Open event to restore settings to their previous state.

Use Syntax 2 to restore settings from an INI file.

Examples

This example calls the of_Restore function to restore settings, as saved in the EISAPP section of the registry:

```
String   ls_appkey
```

```
Integer   li_return

ls_appkey = "HKEY_CURRENT_USER\Software\EISAPP"
li_return = &
   this.inv_preference.of_Restore(ls_appkey)
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Unable to restore settings")
END IF
```

Syntax 2 **To restore settings from an INI file**

Description

Restores window size, position, toolbar settings, and menu item settings from an INI file.

Access

Public

Syntax

instancename.**of_Restore** (*inifile, section*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
inifile	String specifying the INI file from which to restore settings
section	String specifying the INI file section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function in a window Open event to restore settings to their previous state.

Use Syntax 1 to restore settings from the registry.

Examples

This example calls the of_Restore function to restore settings, as saved in the EISAPP section of the registry:

```
String   ls_inifile
Integer  li_return

ls_inifile = gnv_app.of_GetUserINIFile( )
li_return = this.inv_preference.of_Restore &
   (ls_inifile, "Eisapp")
```

of_RestoreMenu

Description

Restores menu items from the registry or an INI file.

Access

Protected

Syntax

instancename.**of_RestoreMenu** (*useregistry*, *keyorinifile*, *section*, *menu*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file
keyorinifile	String specifying either the registry key or the INI file from which to restore settings
section	String specifying the INI file section containing the settings
menu	Menu variable specifying the menu to be restored

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Restore function:

```
...
IF ib_menuitems THEN
   li_rc = of_RestoreMenu(ab_UseRegistry, &
      as_KeyOrIni, as_IniSection, &
         iw_requestor.MenuId)
END IF
...
```

of_RestorePositiveNumber

Description

Accesses the registry or an INI file and reports whether the specified key value returns a positive or a negative number.

Access

Protected

Syntax

instancename.**of_RestorePositiveNumber** (*useregistry*, *keyorinifile*, *section*, *key*, *value*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
keyorinifile	String specifying either the registry key or the INI file from which to restore settings
section	String specifying the INI file section containing the settings
key	String specifying the registry or INI file key value
value	Integer into which the function places the value to be restored

Return value

Boolean. Returns TRUE if *key* contains a positive number and FALSE if it contains a negative number.

Usage

Internal.

Examples

This example is from the of_Restore function:

```
...
IF NOT of_RestorePositiveNumber &
   (ab_UseRegistry, as_KeyOrIni, as_IniSection, &
      ls_tbindex+'.x', li_x) THEN &
         lb_toolbarposition=False
...
```

of_Save

Saves window, toolbar, and menu settings in the registry or an INI file. There are two syntaxes:

To	Use
Save settings in the registry	Syntax 1
Save settings in an INI file	Syntax 2

Other versions of of_Save

There are also two undocumented versions of of_Save, which are for internal use.

Syntax 1 **To save settings in the registry**

Description

Saves window size, position, toolbar settings, and menu item settings in the registry.

Access

Public

Syntax

instancename.**of_Save** (*registrykey*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
registrykey	String specifying the registry key in which to save settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function in a window Close event to save the current settings.

Use Syntax 2 to save settings in an INI file.

Examples

This example calls the of_Save function to save settings, as saved in the EISAPP section of the registry:

```
String   ls_appkey
Integer  li_return

ls_appkey = "HKEY_CURRENT_USER\Software\EISAPP"
li_return = this.inv_preference.of_Save(ls_appkey)
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Unable to save settings")
END IF
```

Syntax 2 **To save settings in an INI file**

Description

Saves window size, position, toolbar settings, and menu item settings in an INI file.

Access

Public

Syntax

instancename.**of_Save** (*inifile, section*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Argument	Description
inifile	String specifying the INI file from which to restore settings
section	String specifying the INI file section containing the settings

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function in a window Close event to save current settings.

Use Syntax 1 to save settings in the registry.

Examples

This example calls the of_Save function to restore settings, as saved in the EISAPP section of the registry:

```
String   ls_inifile
Integer  li_return

ls_inifile = gnv_app.of_GetUserINIFile( )
li_return = this.inv_preference.of_Save &
   (ls_inifile, "Eisapp")
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Unable to save settings")
END IF
```

of_SaveMenu

Description

Saves menu items in either the registry or an INI file.

Access

Protected

Syntax

instancename.**of_SaveMenu** (*useregistry*, *keyorinifile*, *section*, *menu*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
useregistry	Boolean specifying whether to use the registry (TRUE) or an INI file
keyorinifile	String specifying either the registry key or the INI file in which to save settings
section	String specifying the INI file section to contain the settings
menu	Menu variable specifying the menu to be saved

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Save function:

```
...
IF of_SaveMenu(ab_UseRegistry, as_KeyOrIni, &
   as_IniSection, iw_requestor.MenuId) < 0 THEN
      Return -1
END IF
...
```

of_SetMenuItems

Description

Specifies whether the service restores menu items.

Access

Public

Syntax

instancename.**of_SetMenuItems** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores menu items (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMenuItems function:

```
Integer  li_return

li_return = &
   this.inv_preference.of_SetMenuItems(TRUE)
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Error in of_SetMenuItems")
END IF
```

of_SetPosSize

Description

Stores the current position and size of the associated window.

Access

Public

Syntax

instancename.**of_SetPosSize** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)

Return value

Integer. Returns 1 if the function succeeds, 0 if the window state is not Normal!, and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Save function:

```
...
IF iw_requestor.WindowState=Normal! THEN
   of_SetPosSize( )
END IF
...
```

of_SetToolbarItemOrder

Description

Specifies whether the service restores toolbar item order.

Access

Public

Syntax

instancename.**of_SetToolbarItemOrder** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores toolbar item order (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetToolbarItemOrder function:

```
Integer  li_return

li_return = &
   this.inv_preference.of_SetToolbarItemOrder(TRUE)
IF li_return = -1 THEN
   MessageBox("Preference", &
```

```
        "Error in of_SetToolbarItemOrder")
END IF
```

of_SetToolbarItemSpace

Description

Specifies whether the service restores the ToolbarItemSpace property.

Access

Public

Syntax

instancename.**of_SetToolbarItemSpace** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores toolbar item space (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetToolbarItemSpace function:

```
Integer  li_return

li_return = &
   this.inv_preference.of_SetToolbarItemSpace(TRUE)
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Error in of_SetToolbarItemSpace")
END IF
```

of_SetToolbarItemVisible

Description

Specifies whether the service restores the ToolbarItemVisible property.

Access

Public

Syntax

instancename.**of_SetToolbarItemVisible** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores the toolbar item Visible property (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetToolbarItemVisible function:

```
Integer  li_return

li_return = &
   this.inv_preference.of_SetToolbarItemVisible(TRUE)
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Error in of_SetToolbarItemVisible")
END IF
```

of_SetToolbars

Description

Specifies whether the service restores toolbars.

Access

Public

Syntax

instancename.**of_SetToolbars** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores toolbars (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetToolbars function:

```
Integer  li_return

li_return = &
   this.inv_preference.of_SetToolbars(TRUE)
IF li_return = -1 THEN
   MessageBox("Preference", "Error in of_SetToolbars")
END IF
```

of_SetToolbarTitles

Description

Specifies whether the service restores toolbar titles.

Access

Public

Syntax

instancename.**of_SetToolbarTitles** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores toolbar titles (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetToolbarTitles function:

```
Integer  li_return

li_return = &
   this.inv_preference.of_SetToolbarTitles(TRUE)
IF li_return = -1 THEN
   MessageBox("Preference", &
      "Error in of_SetToolbarTitles")
END IF
```

of_SetWindow

Description

Specifies whether the service restores window properties.

Access

Public

Syntax

instancename.**of_SetWindow** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
boolean	Boolean specifying whether the preference service restores window properties (TRUE) or not (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetWindow function:

```
Integer  li_return

li_return = &
```

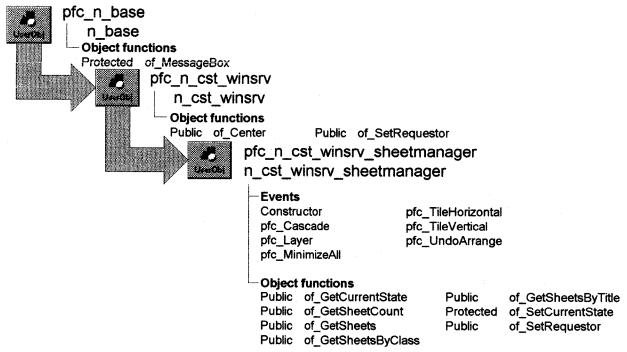
```
    this.inv_preference.of_SetWindow(TRUE)
IF li_return = -1 THEN
    MessageBox("Preference", "Error in of_SetWindow")
END IF
```

n_cst_winsrv_sheetmanager

Description

Sheet management service. The service provides events that are called by PFC menu items to manage multiple sheet display, including the ability to undo a sheet arrangement command. It also provides functions allowing you to access open sheets.

Ancestry



Library

PFCWNSRV.PBL
PFEWNSRV.PBL

Object relationships

w_master

Usage

Enable the sheet management service for all multiple-sheet MDI applications. You don't need this service if your MDI application uses a single-sheet frame.

To use the sheet management service:

1 Create an instance of n_cst_winsrv_sheetmanager by calling the w_frame of_SetSheetManager function.

2 Call n_cst_sheetmanager functions as needed to access information on open sheets.

Structures

Structure	Field	Data type
os_sheet	e_winstate	WindowState
	i_height	Integer
	i_width	Integer
	i_x	Integer
	i_y	Integer
	w_sheetref	Window

See also

n_cst_resize
n_cst_winsrv
n_cst_winsrv_preference
n_cst_winsrv_statusbar
w_master
w_frame

Instance variables

N_cst_winsrv_sheetmanager includes instance variables:

Instance variable	Description	Data type	Access	Usage
ie_arrange	Current arrange type	ArrangeTypes	Protected	Internal
istr_sheet[ ]	Array containing information on the previous state of active sheets	os_sheet	Protected	Internal

Events

N_cst_winsrv_sheetmanager includes precoded events:

Constructor	pfc_TileHorizontal
pfc_Cascade	pfc_TileVertical
pfc_Layer	pfc_UndoArrange
pfc_MinimizeAll	

Constructor

Description

Initializes the ie_arrange instance variable.

Usage

This event executes when the object is created.

pfc_Cascade

Description

Saves information in the istr_sheet array and cascades open sheets.

Usage

This event executes when the user selects Window>Cascade from the menu bar of a menu inherited from m_master.

pfc_Layer

Description — Saves information in the istr_sheet array and layers open sheets.

Usage — This event executes when the user selects Window>Layer from the menu bar of a menu inherited from m_master.

pfc_MinimizeAll

Description — Saves information in the istr_sheet array and minimizes all open sheets.

Usage — This event executes when the user selects Window>Minimize All from the menu bar of a menu inherited from m_master.

pfc_TileHorizontal

Description — Saves information in the istr_sheet array and tiles open sheets horizontally.

Usage — This event executes when the user selects Window>Tile Horizontal from the menu bar of a menu inherited from m_master.

pfc_TileVertical

Description — Saves information in the istr_sheet array and tiles open sheets vertically.

Usage — This event executes when the user selects Window>Tile Vertical from the menu bar of a menu inherited from m_master.

pfc_UndoArrange

Description — Restores all sheets to their previous state and position, using the information in the istr_sheet array.

Usage — This event executes when the user selects Window>Undo Arrange from the menu bar of a menu inherited from m_master.

Functions

N_cst_winsrv_sheetmanager includes precoded object functions:

of_GetCurrentState	of_GetSheetsByClass
of_GetSheetCount	of_GetSheetsByTitle
of_GetSheets	of_SetCurrentState

of_GetCurrentState

Description: Retrieves the current arrange type for undo.

Access: Public

Syntax: *instancename*.**of_GetCurrentState ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)

Return value: ArrangeTypes enumerated variable. Returns the current arrange state of open windows or NULL if Undo Arrange was the last action.

Examples: This example calls the of_GetCurrentState function:

```
ArrangeTypes  lat_arrange

lat_arrange = of_GetCurrentState()
```

of_GetSheetCount

Description: Retrieves the number of sheets open for the current frame.

Access: Public

Syntax: *instancename*.**of_GetSheetCount ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)

Return value: Integer. Returns the number of open sheets for the current frame.

Examples: This example, from a frame window Timer event, calls the of_GetSheetCount function:

```
Integer  li_return

li_return = &
   this.inv_sheetmanager.of_GetSheetCount()
this.SetMicroHelp &
   (String(li_return) + " open sheets")
```

of_GetSheets

Description

Populates an array with a frame's open sheets.

Access

Public

Syntax

instancename.**of_GetSheets** (*sheets*)

Argument	Description
instancename	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
sheets	Array of windows into which the function places the open sheets (passed by reference)

Return value

Integer. Returns the number of open sheets.

Usage

Call this function to determine the number of open sheets and to populate an array containing the handles for all open sheets.

Examples

This example calls the of_GetSheets function:

```
window   lw_sheet, lw_opensheets[ ]
String   ls_sheet
Integer  li_return

ls_sheet = Message.StringParm
OpenSheet(lw_sheet, ls_sheet, this)
li_return = this.inv_sheetmanager.of_GetSheets &
   (lw_opensheets)
this.SetMicroHelp(String(li_return) + " open sheets")
END IF
```

of_GetSheetsByClass

Description

Returns all open sheets of a specified sheet type.

Access

Public

Syntax

instancename.**of_GetSheetsByClass** (*sheets*, *class*)

Argument	Description
instancename	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
sheets	Array of windows into which the function places all sheets of the specified class (passed by reference)
class	String specifying the window class whose open sheets will be returned

Return value

Integer. Returns the number of open sheets of type *class*.

Usage

Call this function to determine the number of open sheets for a particular window class.

Examples

This example calls the of_GetSheetsByClass function to determine an options sheet is open (this assumes a rule that restricts an application to a single open options sheet):

```
window  lw_sheet, lw_opensheets[ ]
String  ls_sheet
Integer li_return

ls_sheet = Message.StringParm
li_return = &
   this.inv_sheetmanager.of_GetSheetsByClass &
      (lw_opensheets, ls_sheet)
IF li_return > 0 THEN
   lw_opensheets[1].WindowState = Normal!
   lw_opensheets[1].BringToTop = TRUE
ELSE
   OpenSheet(lw_sheet, ls_sheet, this)
END IF
```

of_GetSheetsByTitle

Description

Returns all open sheets with a specified title.

Access

Public

Syntax

instancename.**of_GetSheetsByTitle** (*sheets*, *title* {, *partialmatch* })

Argument	Description
instancename	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
sheets	Array of windows into which the function places all sheets with the specified title (passed by reference)
title	String containing the title to be searched for
partialmatch (optional)	Boolean indicating whether the sheet title can contain *title* (TRUE) or must match *title* (FALSE). The default is FALSE

Return value

Integer. Returns the number of open sheets whose titles match the criteria.

Examples

This example, from the w_frame pfc_Open event, calls the of_GetSheetsByTitle function to determine if a sheet is already open for the specified employee:

```
window   lw_sheet, lw_opensheets[ ]
String   ls_sheet, ls_title
Integer  li_return
Long     ll_emp_id

ls_sheet = Message.StringParm
ll_emp_id = Message.LongParm
ls_title = "Employee information for " &
   + String(ll_emp_id)
li_return = &
   this.inv_sheetmanager.of_GetSheetsByTitle &
      (lw_opensheets, ls_title, FALSE)
IF li_return > 0 THEN
   lw_opensheets[1].WindowState = Normal!
   lw_opensheets[1].BringToTop = TRUE
ELSE
   OpenSheetWithParm &
      (lw_sheet, ll_emp_id, ls_sheet, this)
END IF
```

of_SetCurrentState

Description

Sets the current arrange type for undo.

Access

Protected

Syntax

instancename.**of_SetCurrentState** (*arrange*)

Argument	Description
instancename	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
arrange	ArrangeTypes variable specifying the current arrange type

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the pfc_Cascade event:

```
...
of_SetCurrentState(Cascade!)
li_rc = iw_requestor.ArrangeSheets(Cascade!)
Return li_rc
```

n_cst_winsrv_statusbar

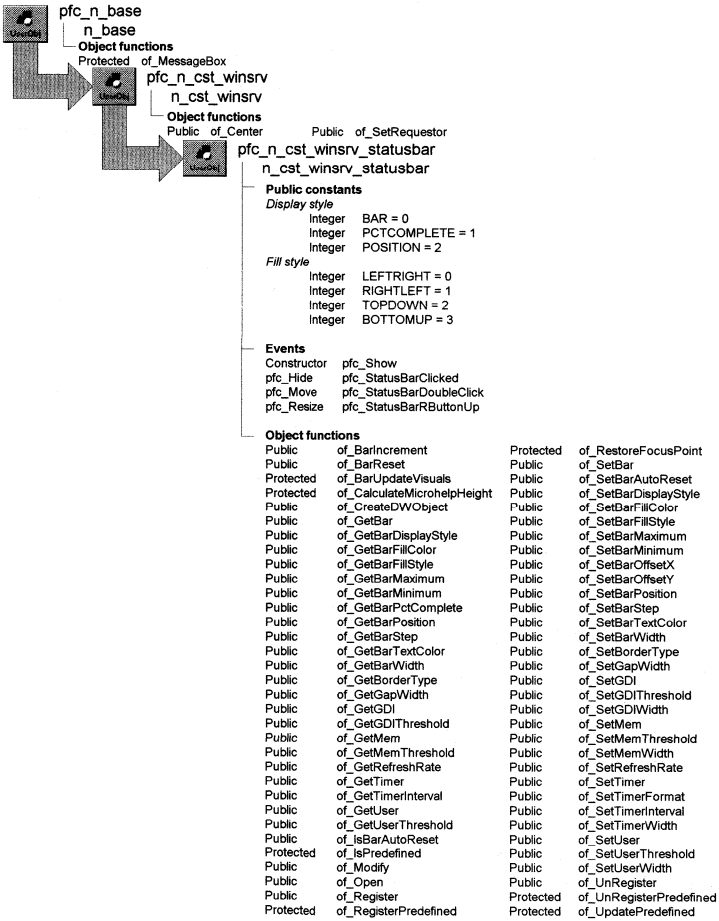
Description

Status bar service. N_cst_winsrv_statusbar provides functions that display memory and time in a frame window's status bar. This service also includes functions that display a progress bar in the status bar area.

On Macintosh The status bar service is not available on Macintosh.

On UNIX The status bar service is not available on UNIX.

Ancestry



Library

PFCWNSRV.PBL
PFEWNSRV.PBL

Usage

Call the functions in this user object to control status bar display.

To use this object:

1 Enable the status bar service by calling the w_frame of_SetStatusBar function:

```
w_frame.of_SetStatusBar(TRUE)
```

2 Call n_cst_winsrv_statusbar functions in the w_frame pfc_PreOpen event to specify the items displayed. The service displays items in the order that their associated functions are called, from left to right. For example, the following example displays memory to the left of the date and time:

```
w_frame.inv_statusbar.of_SetMem(TRUE)
w_frame.inv_statusbar.of_SetTimer(TRUE)
```

3 Call other n_cst_winsrv_statusbar function as appropriate.

Structures

Structure	Field	Data type
os_dwobjects	i_bordertype	Integer
	i_gapwidth	Integer
	i_width	Integer
	s_id	String
	s_type	String
	s_value	String

See also

n_cst_resize
n_cst_winsrv
n_cst_winsrv_preference
n_cst_winsrv_sheetmanager
w_frame
u_progressbar
w_statusbar

Instance variables

Instance variable	Description	Data type	Access	Usage
BAR	Constant set to 0	Integer	Public	Use with of_SetBarDisplayStyle
BITMAP	Constant set to "bitmap"	String	Protected	Internal
BOTTOMUP	Constant set to 3	Integer	Public	Use with of_SetFillStyle
ib_bar	Controls progress bar display	Boolean	Protected	Set with of_SetBar
ib_barautoreset	Controls progress bar autoreset	Boolean	Protected	Set with of_SetBarAutoReset (default is TRUE)
ib_gdi	Indicates whether GDI is displayed	Boolean	Protected	Set with of_SetGDI
ib_mem	Indicates whether memory is displayed	Boolean	Protected	Set with of_SetMem
ib_timer	Indicates whether time is displayed	Boolean	Protected	Set with of_SetTimer
ib_user	Indicates whether user memory is displayed	Boolean	Protected	Set with of_SetUser
ienv_object	Environment object	Environment	Protected	Internal
ii_bardisplaystyle	Bar display style	Integer	Protected	Set with of_SetBarDisplayStyle (default is BAR)
ii_barfillstyle	Bar fill style	Integer	Protected	Set with of_SetBarFillStyle (default is LEFTRIGHT)
ii_barheight	Height	Integer	Protected	Internal
ii_barmax	Progress bar maximum	Integer	Protected	Set with of_SetBarMaximum (default is 100)
ii_barmin	Progress bar minimum	Integer	Protected	Set with of_SetBarMinimum (default is 1)

Instance variable	Description	Data type	Access	Usage
ii_baroffsetx	Bar offset	Integer	Protected	Set with of_SetBarOffsetX (default is 10)
ii_baroffsety	Bar offset	Integer	Protected	Set with of_SetBarOffsetY (default is 7)
ii_barpctcomplete	Current progress	Integer	Protected	Internal
ii_barposition	Progress bar position	Integer	Protected	Set with of_SetBarPosition. Default is 0
ii_barstartx	Progress bar start x	Integer	Protected	Internal
ii_barstarty	Progress bar start y	Integer	Protected	Internal
ii_barstep	Progress bar step	Integer	Protected	Set with of_SetBarStep (default is 10)
ii_barwidth	Progress bar width	Integer	Protected	Set with of_SetBarWidth (default is 200)
ii_bordertype	Border type	Integer	Protected	Set with of_SetBorderType (default is 5)
ii_gapwidth	Gap width	Integer	Protected	Set with of_SetGapWidth (default is 15)
ii_gdiwidth	Width of GDI item	Integer	Protected	Set with of_SetGDIWidth (default is 250)
ii_memwidth	Width of memory item	Integer	Protected	Set with of_SetMemWidth (default is 350)
ii_refreshrate	Number of seconds between refreshes	Integer	Protected	Set with of_SetRefreshRate (default is 60)
ii_timerwidth	Width of timer item	Integer	Protected	Set with of_SetTimerWidth (default is 400)
ii_userwidth	Width of user memory item	Integer	Protected	Set with of_SetUserWidth (default is 270)

Instance variable	Description	Data type	Access	Usage
il_barfillcolor	Progress bar fill color	Long	Protected	Set with of_SetBarFillColor (default is 10789024 - medium gray)
il_bartextcolor	Progress bar text	Long	Protected	Set with of_SetBarTextColor (default is 0 - black)
il_gdithreshold	Threshold under which a warning displays	Long	Protected	Set with of_SetGDIThreshold
il_memthreshold	Threshold under which a warning displays	Long	Protected	Set with of_SetMemThreshold
il_timerinterval	In milliseconds	Long	Protected	Set with of_SetTimerInterval (default is 1000)
il_userthreshold	Threshold under which a warning displays	Long	Protected	Set with of_SetUserThreshold
is_timerformat	Timer format	String	Protected	Set with of_SetTimerFormat (default is m-d-yy h:mm:ss)
istr_dwobjects[ ]	Status bar items	os_dwobjects	Protected	Internal
iw_statusbar	Status bar window	w_statusbar	Protected	Internal
LEFTRIGHT	Constant set to 0	Integer	Public	Use with of_SetBarFillStyle
PCTCOMPLETE	Constant set to 1	Integer	Public	Use with of_SetBarStyle
POSITION	Constant set to 2	Integer	Public	Use with of_SetBarDisplayStyle
PREDEFINED	Constant set to "predefined"	String	Protected	Internal
RIGHTLEFT	Constant set to 1	Integer	Public	Use with of_SetBarFillStyle
TEXT	Constant set to "text"	String	Protected	Internal

Instance variable	Description	Data type	Access	Usage
TOPDOWN	Constant set to 2	Integer	Public	Use with of_SetBarFillStyle

Events

N_cst_winsrv_statusbar includes precoded event scripts:

Constructor	pfc_Show
pfc_Hide	pfc_StatusBarClicked
pfc_Move	pfc_StatusBarDoubleClick
pfc_Resize	pfc_StatusBarRButtonUp

Constructor

Description — Populates the ienv_environment instance variable.

Usage — This event executes when the object is created.

pfc_Hide

Description — Hides the w_statusbar window when the frame window is hidden.

Return value — Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage — The w_frame Hide event calls this event.

pfc_Move

Description — Moves and resizes the w_statusbar window to match the new position and size of the associated frame window.

Syntax — *instancename.Event* **pfc_Move** (*x*, *y*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar
x	Integer specifying the new x position of the associated frame window. From within the pfc_Move event, access this value through the *ai_xpos* argument

Argument	Description
y	Integer specifying the new y position of the associated frame window. From within the pfc_Move event, access this value through the *ai_ypos* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The w_frame Move event calls this event.

pfc_Resize

Description

Resizes the w_statusbar window based on the new size and position of the associated frame window.

Syntax

instancename.Event **pfc_Resize** (*sizetype, newwidth, newheight*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar
sizetype	Unsigned Long specifying the type of resize. Values are: ♦ **0** The frame window was resized but was not maximized or minimized ♦ **1** The frame window was minimized ♦ **2** The frame window was maximized From within the pfc_Resize event, access this value through the *aul_sizetype* argument
newwidth	Integer specifying the new window width, in PBUs
newheight	Integer specifying the new window height, in PBUs

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The w_master Resize event calls this event.

pfc_Show

Description

Displays the w_statusbar window.

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

The w_master Show event calls this event.

pfc_StatusBarClicked

Description

Empty user event into which you place code to execute when the user clicks in the status bar window.

Syntax

instancename.Event **pfc_StatusBarClicked** (*x, y, name*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar
x	Integer specifying the x position of the mouse click. From within the pfc_StatusBarClicked event, access this value through the *ai_xpos* argument
y	Integer specifying the x position of the mouse click. From within the pfc_StatusBarClicked event, access this value through the *ai_ypos* argument
name	String containing the name of the status bar element that was clicked (for example, this argument contains pfc_mem if the user clicks in the memory display). From within the pfc_StatusBarClicked event, access this value through the *as_name* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Examples

This example shows code you might add to the pfc_StatusBarClicked event to display a popup calculator window. This assumes that you have called the of_Register function to display text for the calculator on the status bar and that you have created a window named w_popcalc:

```
IF as_name = "calc" THEN
   Open(w_popcalc, iw_requestor)
END IF
```

pfc_StatusBarDoubleClick

Description

Empty user event into which you place code to execute when the user double-clicks in the status bar window.

Syntax

instancename.Event **pfc_StatusBarDoubleClick** (*x, y, name*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar

Argument	Description
x	Integer specifying the x position of the mouse click. From within the pfc_StatusBarDoubleClick event, access this value through the *ai_xpos* argument
y	Integer specifying the x position of the mouse click. From within the pfc_StatusBarDoubleClick event, access this value through the *ai_ypos* argument
name	String containing the name of the status bar element that was clicked (for example, this argument contains pfc_mem if the user clicks in the memory display). From within the pfc_StatusBarDoubleClick event, access this value through the *as_name* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Usage

Add code to this event to perform some action when the user double-clicks in the status bar window.

Examples

This example shows code you might add to the pfc_StatusBarDoubleClick event to display a popup calculator window. This assumes that you have called the of_Register function to display text for the calculator on the status bar and that you have created a window named w_popcalc:

```
IF as_name = "calc" THEN
   Open(w_popcalc, iw_requestor)
END IF
```

pfc_StatusBarRButtonUp

Description

Empty user event into which you place code to execute when the user right-clicks in the status bar window.

Syntax

instancename.Event **pfc_StatusBarRButtonUp** (*x*, *y*, *name*)

Argument	Description
instancename	Instance name of n_cst_rtefind
x	Integer specifying the x position of the mouse click. From within the pfc_StatusBarRButtonUp event, access this value through the *ai_xpos* argument
y	Integer specifying the x position of the mouse click. From within the pfc_StatusBarRButtonUp event, access this value through the *ai_ypos* argument

Argument	Description
name	String containing the name of the status bar element that was clicked (for example, this argument contains pfc_mem if the user clicks in the memory display). From within the pfc_StatusBarRButtonUp event, access this value through the *as_name* argument

Return value

Integer. Returns 1 if the event succeeds and -1 if an error occurs.

Examples

This example shows code you might add to the pfc_StatusBarRButtonUp event to display a popup calculator window. This assumes that you have called the of_Register function to display text for the calculator on the status bar and that you have created a window named w_popcalc:

```
IF as_name = "calc" THEN
   Open(w_popcalc, iw_requestor)
END IF
```

Functions

N_cst_winsrv_statusbar includes precoded object functions:

of_BarIncrement	of_RestoreFocusPoint
of_BarReset	of_SetBar
of_BarUpdateVisuals	of_SetBarAutoReset
of_CalculateMicrohelpHeight	of_SetBarDisplayStyle
of_CreateDWObject	of_SetBarFillColor
of_GetBar	of_SetBarFillStyle
of_GetBarDisplayStyle	of_SetBarMaximum
of_GetBarFillColor	of_SetBarMinimum
of_GetBarFillStyle	of_SetBarOffsetX
of_GetBarMaximum	of_SetBarOffsetY
of_GetBarMinimum	of_SetBarPosition
of_GetBarPctComplete	of_SetBarStep
of_GetBarPosition	of_SetBarTextColor
of_GetBarStep	of_SetBarWidth
of_GetBarTextColor	of_SetBorderType
of_GetBarWidth	of_SetGapWidth
of_GetBorderType	of_SetGDI
of_GetGapWidth	of_SetGDIThreshold
of_GetGDI	of_SetGDIWidth
of_GetGDIThreshold	of_SetMem
of_GetMem	of_SetMemThreshold

of_GetMemThreshold	of_SetMemWidth
of_GetRefreshRate	of_SetRefreshRate
of_GetTimer	of_SetTimer
of_GetTimerInterval	of_SetTimerFormat
of_GetUser	of_SetTimerInterval
of_GetUserThreshold	of_SetTimerWidth
of_IsBarAutoReset	of_SetUser
of_IsPredefined	of_SetUserThreshold
of_Modify	of_SetUserWidth
of_Open	of_UnRegister
of_Register	of_UnRegisterPredefined
of_RegisterPredefined	of_UpdatePredefined

of_BarIncrement

Description

Increments the progress by either the default increment value or by a specified value.

Access

Public

Syntax

instancename.**of_BarIncrement** ({ *incrementvalue* })

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
incrementvalue (optional)	Integer specifying the increment value. The default increment value is 10, which you can modify by calling the of_SetBarStep function

Return value

Integer. Returns the current percent complete.

Examples

This example calls the of_BarIncrement function:

```
Integer  li_count
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBar(TRUE)
lw_frame.inv_statusbar.of_SetBarPosition(0)
For li_count = 1 to 100 STEP 1
   lw_frame.inv_statusbar.of_BarIncrement(1)
NEXT
lw_frame.inv_statusbar.of_SetBar(FALSE)
```

of_BarReset

Description

Clears the progress bar displayed in the status bar.

Access

Public

Syntax

instancename.**of_BarReset** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the of_SetBarPosition function:

```
...
if ldc_completion >= 1 and ib_barautoreset then
   this.post function ()
end if
...
```

of_BarUpdateVisuals

Description

Creates a string used to resize the progress bar and update the display text.

Access

Protected

Syntax

instancename.**of_BarUpdateVisuals** (*completionvalue*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
completionvalue	Decimal specifying the completion value of the progress bar

Return value

String. Returns the Modify string used to set the progress bar if the function succeeds and an empty string if an error occurs.

Usage

Internal.

Examples

This example is from the of_SetBarPosition function:

```
...
ls_modifyexp = of_BarUpdateVisuals(ldc_completion)
```

```
li_rc = of_Modify(ls_visible + ls_modifyexp)
...
```

of_CalculateMicrohelpHeight

Description: Retrieves the environment-specific height of the MicroHelp area.

Access: Protected

Syntax: *instancename*.**of_CalculateMicrohelpHeight ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage: Internal.

Examples: This example is from the of_CreateDWObject function:

```
...
li_microhelpheight = of_CalculateMicroHelpHeight()
...
```

of_CreateDWObject

Description: Creates and validates the DataWindow syntax used to create the status bar.

Access: Public

Syntax: *instancename*.**of_CreateDWObject (*createstatement, width, height*)**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
createstatement	String into which the function places syntax for the DataWindow Create statement (passed by reference)
width	Integer into which the function places the width of the created object (passed by reference)
height	Integer into which the function places the height of the created object (passed by reference)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to create and validate the string used to create a visual element on the status bar.

Examples

This example is from the w_statusbar of_CreateVisuals function:

```
...
li_rc = &
   iw_parentwindow.inv_statusbar.of_CreateDWObject &
      (ls_dwdefinition, ii_winmaxwidth, li_height)
...
```

of_GetBar

Description

Reports whether the status bar includes a progress bar.

Access

Public

Syntax

instancename.**of_GetBar** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Boolean. Returns TRUE if the status bar includes a progress bar and FALSE if it does not.

Examples

This example calls the of_GetBar function:

```
n_cst_conversion   lnv_conversion
Boolean   lb_progressbar
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
lb_progressbar = &
   lw_frame.inv_statusbar.of_GetBar()
MessageBox("Status Bar", &
   "Progress bar is " &
      + lnv_conversion.of_String(lb_progressbar))
```

of_GetBarDisplayStyle

Description

Retrieves the progress bar display style. The display style specifies the information that displays within the bar while tracking progress.

Access

Public

Syntax

instancename.**of_GetBarDisplayStyle** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns values as follows:

- **0 (BAR)** Progress bar only
- **1 (PCTCOMPLETE)** Progress bar and percent complete, including the percent sign
- **2 (POSITION)** Progress bar and percent complete with no percent sign

Examples

This example calls the of_GetBarDisplayStyle function:

```
Integer   li_style
String    ls_style[3] = &
   {"Bar", "% Complete", "Complete"}
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
li_style =  &
  lw_frame.inv_statusbar.of_GetBarDisplayStyle()
MessageBox("Progress Bar", &
   "Bar Style value is " &
      + String(li_style) + "-" + &
         ls_style[li_style + 1])
```

of_GetBarFillColor

Description

Retrieves the progress bar's current color.

Access

Public

Syntax

instancename.**of_GetBarFillColor** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Long. Returns the progress bar's fill color.

Examples

This example calls the of_GetBarFillColor function:

```
Long   ll_color
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
ll_color = lw_frame.inv_statusbar.of_GetBarFillColor()
MessageBox("Progress Bar", &
    "Fill color property is " &
       + String(ll_color))
```

of_GetBarFillStyle

Description

Retrieves the progress bar's fill style.

Access

Public

Syntax

instancename.**of_GetBarFillStyle ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns values as follows:

- **0 (LEFTRIGHT)** The progress bar fills from left to right
- **1 (RIGHTLEFT)** The progress bar fills from right to left
- **2 (TOPDOWN)** The progress bar fills from top to bottom
- **3 (BOTTOMUP)** The progress bar fills from bottom to top

Examples

This example calls the of_GetBarFillStyle function:

```
Integer  li_fillstyle
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
```

```
li_fillstyle = &
   lw_frame.inv_statusbar.of_GetBarFillStyle()
MessageBox("Progress Bar", &
   "Fill style is " + String(li_fillstyle))
```

of_GetBarMaximum

Description: Retrieves the maximum increment value used by the progress bar.

Access: Public

Syntax: *instancename*.**of_GetBarMaximum ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value: Integer. Returns the maximum increment value for the progress bar.

Examples: This example calls the of_GetBarMaximum function:

```
Integer   li_max
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
li_max = lw_frame.inv_statusbar.of_GetBarMaximum()
MessageBox("Progress Bar", &
   "Max value is " &
      + String(li_max))
```

of_GetBarMinimum

Description: Retrieves the minimum value from which incrementation starts.

Access: Public

Syntax: *instancename*.**of_GetBarMinimum ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value: Integer. Returns the minimum increment value.

Examples

This example calls the of_GetBarMinimum function:

```
Integer   li_min
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
li_max = lw_frame.inv_statusbar.of_GetBarMinimum()
MessageBox("Progress Bar", &
    "Min value is " + String(li_min))
```

of_GetBarPctComplete

Description

Retrieves the progress bar's current percent complete.

Access

Public

Syntax

instancename.**of_GetBarPctComplete ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the current percent complete for the progress bar.

Examples

This example calls the of_GetBarPctComplete function:

```
Integer   li_count, li_current
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBar(TRUE)
lw_frame.inv_statusbar.of_SetBarPosition(0)
For li_count = 1 to 100 STEP 1
    lw_frame.inv_statusbar.of_BarIncrement(1)
    li_current = &
       lw_frame.inv_statusbar.of_GetBarPctComplete()
    lw_frame.SetMicroHelp(String(li_current) + "%")
NEXT
lw_frame.inv_statusbar.of_SetBar(FALSE)
```

of_GetBarPosition

Description

Retrieves the progress bar's current increment value.

Access

Public

Syntax

instancename.**of_GetBarPosition ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the current increment value for the progress bar.

Examples

This example calls the of_GetBarPosition function:

```
Integer   li_count, li_current
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBar(TRUE)
lw_frame.inv_statusbar.of_SetBarPosition(0)
For li_count = 1 to 100 STEP 1
   lw_frame.inv_statusbar.of_BarIncrement(1)
   li_current = &
      lw_frame.inv_statusbar.of_GetBarPosition()
   lw_frame.SetMicroHelp(String(li_current) + "%")
NEXT
lw_frame.inv_statusbar.of_SetBar(FALSE)
```

of_GetBarStep

Description

Retrieves the step value that the progress bar uses during a default increment process.

Access

Public

Syntax

instancename.**of_GetBarStep ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the step value.

Examples

This example calls the of_GetBarStep function:

```
Integer  li_step
w_frame  lw_frame
```

```
lw_frame = gnv_app.of_GetFrame()
li_step = lw_frame.inv_statusbar.of_GetBarStep()
MessageBox("Progress Bar", &
    "Step value is " &
       + String(li_step))
```

of_GetBarTextColor

Description

Retrieves the progress bar text color.

Access

Public

Syntax

instancename.**of_GetBarTextColor ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Long. Returns the progress bar text color.

Examples

This example calls the of_GetBarTextColor function:

```
Long   ll_color
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
ll_color = lw_frame.inv_statusbar.of_GetBarTextColor()
MessageBox("Progress Bar", &
    "Text color property is " + String(ll_color))
```

of_GetBarWidth

Description

Retrieves the width of the progress bar area within the status bar.

Access

Public

Syntax

instancename.**of_GetBarWidth ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the width of the progress bar area.

Examples

This example calls the of_GetBarWidth function:

```
Integer   li_width
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
li_width = &
   lw_frame.inv_statusbar.of_GetBarWidth()
MessageBox("Status Bar", &
   "Progress bar width is " + String(li_width))
```

of_GetBorderType

Description

Retrieves the default border type for new objects on the status bar.

Access

Public

Syntax

instancename.**of_GetBorderType** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the border type used when creating new objects on the status bar:

- **0** None
- **1** Shadow
- **2** Rectangle
- **3** Resize
- **4** Line
- **5** 3D lowered
- **6** 3D raised

Examples

This example calls the of_GetBorderType function:

```
Integer  li_border
w_frame  lw_frame
String   ls_borderstyle[] = &
   {"None", "Shadow", "Rect", "Resize", &
```

```
        "Line", "3D Lowered", "3D Raised" }

lw_frame = gnv_app.of_GetFrame()
li_border = &
    lw_frame.inv_statusbar.of_GetBorderType()
MessageBox("Status Bar", &
    "Progress bar border is " &
       + ls_borderstyle[li_border + 1])
```

of_GetGapWidth

Description

Retrieves the default between-object spacing in PBUs for new objects on the status bar.

Access

Public

Syntax

instancename.**of_GetGapWidth ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the default between-object spacing used when creating new objects on the status bar.

Examples

This example calls the of_GetGapWidth function:

```
Integer  li_gapwidth
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_gapwidth = &
    lw_frame.inv_statusbar.of_GetGapWidth()
MessageBox("Status Bar", &
    "Gap width is " + String(li_gapwidth))
```

of_GetGDI

Description

Indicates whether GDI memory is displayed in the status bar.

Access

Public

Syntax

instancename.**of_GetGDI ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Boolean. Returns TRUE if GDI memory is displayed and FALSE if it is not.

Usage

GDI display is not available in 32-bit Windows.

Examples

This example calls the of_GetGDI function:

```
n_cst_conversion  lnv_conversion
Boolean  lb_gdi
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lb_gdi = &
   lw_frame.inv_statusbar.of_GetGDI()
MessageBox("Status Bar", &
   "GDI display is " &
      + lnv_conversion.of_String(lb_gdi))
```

of_GetGDIThreshold

Description

Retrieves the current GDI threshold value.

Access

Public

Syntax

instancename.**of_GetGDIThreshold ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Long. Returns the current GDI threshold value.

Usage

GDI display is not available in 32-bit Windows.

Examples

This example calls the of_GetGDIThreshold function:

```
Integer  li_threshold
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_threshold = &
```

```
    lw_frame.inv_statusbar.of_GetGDIThreshold()
MessageBox("Status Bar", &
    "GDI threshold is " &
       + String(li_threshold))
```

of_GetMem

Description — Indicates whether free memory is displayed in the status bar.

Access — Public

Syntax — *instancename*.**of_GetMem ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value — Boolean. Returns TRUE if free memory is displayed and FALSE if it is not.

Examples — This example calls the of_GetMem function:

```
n_cst_conversion  lnv_conversion
Boolean  lb_mem
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lb_mem = &
    lw_frame.inv_statusbar.of_GetMem()
MessageBox("Status Bar", &
    "Free memory display is " &
       + lnv_conversion.of_String(lb_mem))
```

of_GetMemThreshold

Description — Retrieves the current free memory threshold value.

Access — Public

Syntax — *instancename*.**of_GetMemThreshold ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Long. Returns the current free memory threshold value.

Examples

This example calls the of_GetMemThreshold function:

```
Integer  li_threshold
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_threshold = &
   lw_frame.inv_statusbar.of_GetMemThreshold()
MessageBox("Status Bar", &
   "Free memory threshold is " &
      + String(li_threshold))
```

of_GetRefreshRate

Description

Retrieves the current refresh rate for all objects but the timer.

Access

Public

Syntax

instancename.**of_GetRefreshRate ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Integer. Returns the current refresh rate.

Examples

This example calls the of_GetRefreshRate function:

```
Integer  li_refresh
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_refresh = &
   lw_frame.inv_statusbar.of_GetRefreshRate()
MessageBox("Status Bar", &
   "Refresh rate is " &
      + String(li_refresh))
```

of_GetTimer

Description

Reports whether the time and date is displayed in the status bar.

Access

Public

Syntax

instancename.**of_GetTimer ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Boolean. Returns TRUE if time and date are displayed and FALSE if they are not.

Examples

This example calls the of_GetTimer function:

```
n_cst_conversion  lnv_conversion
Boolean  lb_timer
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lb_timer = &
   lw_frame.inv_statusbar.of_GetTimer()
MessageBox("Status Bar", &
   "Timer display is " &
      + lnv_conversion.of_String(lb_timer))
```

of_GetTimerInterval

Description

Retrieves the refresh rate for the timer object.

Access

Public

Syntax

instancename.**of_GetTimerInterval ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Long. Returns the number of milliseconds between timer refreshes.

Usage

This function applies to the timer object only. To determine the refresh rate for free memory, GDI memory, and user memory, call the of_GetRefreshRate function.

Examples

This example calls the of_GetTimerInterval function:

```
Integer  li_interval
```

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_interval = &
   lw_frame.inv_statusbar.of_GetTimerInterval()
MessageBox("Status Bar", &
   "Timer interval is " &
      + String(li_interval))
```

of_GetUser

Description

Reports whether the user memory is displayed in the status bar.

Access

Public

Syntax

instancename.**of_GetUser** ( )

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Boolean. Returns TRUE if user memory is displayed and FALSE if it is not.

Usage

User memory display is not available in 32-bit Windows.

Examples

This example calls the of_GetUser function:

```
n_cst_conversion   lnv_conversion
Boolean  lb_user
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lb_user = &
   lw_frame.inv_statusbar.of_GetUser()
MessageBox("Status Bar", &
   "User memory display is " &
      + lnv_conversion.of_String(lb_user))
```

of_GetUserThreshold

Description

Retrieves the current user memory threshold value.

Access

Public

Syntax

instancename.**of_GetUserThreshold ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Long. Returns the current user memory threshold value.

Usage

User memory display is not available in 32-bit Windows.

Examples

This example calls the of_GetUserThreshold function:

```
Integer  li_threshold
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_threshold = &
   lw_frame.inv_statusbar.of_GetUserThreshold()
MessageBox("Status Bar", &
   "User memory threshold is " &
      + String(li_threshold))
```

of_IsBarAutoReset

Description

Reports whether the progress bar returns to zero after it reaches the maximum value.

Access

Public

Syntax

instancename.**of_IsBarAutoReset ( )**

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Return value

Boolean. Returns TRUE if n_cst_winsrv_statusbar clears the progress bar when it reaches 100% and FALSE if it does not.

Examples

This example calls the of_IsBarAutoReset function:

```
n_cst_conversion  lnv_conversion
Boolean  lb_autoclear
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
```

```
lb_autoclear = &
   lw_frame.inv_statusbar.of_IsBarAutoReset()
MessageBox("Progress Bar", &
   "Autoclear property is " &
      + lnv_conversion.of_String(lb_autoclear))
```

of_IsPredefined

Description

Determines whether the specified argument is a predefined ID. N_cst_winsrv_statusbar uses predefined IDs for GDI memory, free memory, user memory, the time, and the progress bar.

Access

Protected

Syntax

instancename.**of_IsPredefined** (*id*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	String containing the ID to be tested

Return value

Boolean. Returns TRUE if *id* is one of the predefined IDs and FALSE if it is not.

Usage

Internal.

Examples

This example is from the of_RegisterPredefined function:

```
...
IF NOT of_IsPredefined(Trim(as_id)) THEN
   Return -1
END IF
...
```

of_Modify

Description

Modifies the display of a user-defined status bar object.

Access

Public

Syntax

instancename.**of_Modify** (*id, newvalue*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	Either of the following: ♦ String containing the identifier specified in the of_Register function that created the status bar object ♦ String containing a modify expression to be used on the modification of the object
newvalue	String containing either the new text or a bitmap filename. Do not specify this argument if *id* contains a modify expression

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example is from the of_SetBarPosition function:

```
...
ls_modifyexp = of_BarUpdateVisuals(ldc_completion)
li_rc = of_Modify(ls_visible + ls_modifyexp)
...
```

of_Open

Description

Opens or closes the w_statusbar window.

Access

Public

Syntax

instancename.**of_Open** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
boolean	Boolean indicating whether to open (TRUE) or close (FALSE) the w_statusbar window

Return value

Integer. Returns 1 if the function succeeds, 0 if no action was taken, and -1 if an error occurs.

Examples

This example is from the w_frame pfc_PostOpen event:

```
IF IsValid (inv_statusbar) THEN
   inv_statusbar.of_Open(TRUE)
END IF
```

of_Register

Description

Registers and displays a new status bar object.

Access

Public

Syntax

instancename.**of_Register** (*id*, *type*, *value*, *width* {, *bordertype* {, *gapwidth* } })

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	String specifying a unique identifier for the status bar object
type	String specifying whether the object displays text or a bitmap. Valid values are: ♦ Text ♦ Bitmap
value	String specifying either the display text or bitmap name for the object
width	Integer specifying object width, in PBUs
bordertype (optional)	Integer specifying the border type. Values are: ♦ 0 None ♦ 1 Shadow ♦ 2 Rectangle ♦ 3 Resize ♦ 4 Line ♦ 5 3D Lowered (default) ♦ 6 3D Raised
gapwidth (optional)	Integer specifying the number of PBUs between this object and other objects. The default is 15

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example from a frame Open event calls the of_Register function to enable a popup calculator:

```
...
this.inv_statusbar.of_Register &
   ("calc", "text", "Calculator", 300)
...
```

of_RegisterPredefined

Description

Registers one of the predefined objects:

Timer
GDI memory
User memory
Free memory
Progress bar

Access

Protected

Syntax

instancename.**of_RegisterPredefined** (*id, width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	String specifying one of the following: ♦ **Pfc_timer** Time and date ♦ **Pfc_gdi** GDI memory ♦ **Pfc_user** User memory ♦ **Pfc_mem** Free memory ♦ **Pfc_progress** Progress bar
width	Integer specifying the object's width, in PBUs

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SetUser function:

```
...
IF ib_user THEN
    li_rc = of_RegisterPreDefined &
       ('pfc_user', ii_userwidth)
ELSE
    li_rc = of_UnRegisterPreDefined('pfc_user')
END IF
Return li_rc
```

of_RestoreFocusPoint

Description

Sets focus to the specified control.

Access

Protected

Syntax

instancename.**of_RestoreFocusPoint** (*control*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
control	GraphicObject referencing the control to which focus is set. If *control* is invalid, the function returns focus to the parent window

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_Open function:

```
...
IF ab_switch THEN
   lgo_withfocus = GetFocus()
   li_rc = Open(iw_statusbar, iw_requestor)
   of_RestoreFocusPoint(lgo_withfocus)
   IF IsValid(iw_statusbar) THEN
      iw_statusbar.of_SetRefreshRate(ii_refreshrate)
   END IF
...
```

of_SetBar

Description

Enables the display of the progress bar in the status bar.

Access

Public

Syntax

instancename.**of_SetBar** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Argument	Description
boolean	Boolean specifying whether to display the progress bar in the status bar (TRUE) or remove the progress bar from the status bar (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBar function:

```
Integer  li_count, li_current
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBar(TRUE)
lw_frame.inv_statusbar.of_SetBarPosition(0)
For li_count = 1 to 100 STEP 1
   lw_frame.inv_statusbar.of_BarIncrement(1)
   li_current = &
      lw_frame.inv_statusbar.of_GetBarPosition()
   lw_frame.SetMicroHelp(String(li_current) + "%")
NEXT
lw_frame.inv_statusbar.of_SetBar(FALSE)
```

of_SetBarAutoReset

Description

Specifies whether to clear the progress bar when it reaches 100%.

Access

Public

Syntax

instancename.**of_SetBarAutoReset** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
boolean	Boolean specifying whether to clear the status bar when it reaches 100% (TRUE) or leave it visible (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarAutoReset function:

```
w_frame  lw_frame
```

```
lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBarAutoReset(FALSE)
```

of_SetBarDisplayStyle

Description: Establishes a display style for the progress bar.

Access: Public

Syntax: *instancename*.**of_SetBarDisplayStyle** (*displaystyle*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
displaystyle	Integer or n_cst_winsrv_statusbar constant specifying the progress bar display style: ♦ **0 or BAR (default)** Bar only, no text ♦ **1 or PCTCOMPLETE** Bar displays percent complete, including the percent sign ♦ **2 or POSITION** Bar displays percent complete with no percent sign

Return value: Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples: This example calls the of_SetBarDisplayStyle function:

```
...
this.inv_statusbar.of_Register &
   ("calc", "text", "Calculator", 300)
this.inv_statusbar.of_SetBar(TRUE)
this.inv_statusbar.of_SetTimer(TRUE)
this.inv_statusbar.of_SetBarDisplayStyle &
   (this.inv_statusbar.PCTCOMPLETE)
...
```

of_SetBarFillColor

Description: Specifies the progress bar color.

Access: Public

Syntax: *instancename*.**of_SetBarFillColor** (*color*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
color	Long specifying progress bar color

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarFillColor function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBarFillColor &
   (RGB(255, 0, 0))
```

of_SetBarFillStyle

Description

Establishes a fill style for the progress bar.

Access

Public

Syntax

instancename.**of_SetBarFillStyle** (*style*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar
style	Integer or n_cst_winsrv_statusbar constant specifying the fill style: ♦ **0 or TOPDOWN** The bar fills from top to bottom ♦ **1 or BOTTOMUP** The bar fills from bottom to top ♦ **2 or LEFTRIGHT (default)** The bar fills from left to right ♦ **3 or RIGHTLEFT** The bar fills from right to left

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarFillStyle function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBarFillStyle &
   (lw_frame.inv_statusbar.RIGHTLEFT)
```

of_SetBarMaximum

Description

Sets the maximum increment value. The progress bar uses this value as the basis for its percent complete calculation.

Access

Public

Syntax

instancename.**of_SetBarMaximum** (*maximum*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
maximum	Integer specifying the maximum increment value (the default is 100)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The of_BarIncrement function uses this value as the dividend in its percent complete calculation (percent complete = current position/maximum).

Examples

This example calls the of_SetBarMaximum function:

```
...
lw_frame = gnv_app.of_GetFrame()
li_max = 80
lw_frame.inv_statusbar.of_SetBarMaximum(li_max)
...
```

of_SetBarMinimum

Description

Sets the maximum increment value. The progress bar uses this value as the basis for its percent complete calculation.

Access

Public

Syntax

instancename.**of_SetBarMinimum** (*maximum*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
maximum	Integer specifying the maximum increment value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The default for this value is 100. The of_BarIncrement function uses this value as the dividend in its percent complete calculation (percent complete = current position/maximum).

Examples

This example calls the of_SetBarMinimum function:

```
...
lw_frame = gnv_app.of_GetFrame()
li_max = 80
lw_frame.inv_statusbar.of_SetBarMinimum(li_max)
...
```

of_SetBarOffsetX

Description

Specifies a new *x* offset for the progress bar.

Access

Public

Syntax

instancename.**of_SetBarOffsetX** (*offset*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
offset	Integer specifying an *x* offset for the progress bar, in PBUs (the default is 10 PBUs)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarOffsetX function:

```
...
this.inv_statusbar.of_SetBarOffSetX(20)
this.inv_statusbar.of_SetBarOffSetY(15)
...
```

of_SetBarOffsetY

Description

Specifies a new *y* offset for the progress bar.

Access

Public

Syntax

instancename.**of_SetBarOffsetY** (*offset*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
offset	Integer specifying a *y* offset for the progress bar, in PBUs (the default is 7 PBUs)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarOffsetY function:

```
...
this.inv_statusbar.of_SetBarOffSetX(20)
this.inv_statusbar.of_SetBarOffSetY(15)
...
```

of_SetBarPosition

Description

Specifies the progress bar's current position.

Access

Public

Syntax

instancename.**of_SetBarPosition** (*position*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
position	Integer specifying the progress bar position

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Call this function to establish an initial position for the progress bar before beginning an incrementation process. If *position* is less than 0, this function sets the position to the minimum. If *position* is greater than the maximum (the default maximum is 100), this function sets the position to the maximum. You modify the default maximum by calling the of_SetBarMaximumimum function.

Examples

This example calls the of_SetBarPosition function:

```
Integer  li_count
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
```

```
lw_frame.inv_statusbar.of_SetBarPosition(0)
For li_count = 1 to 100 STEP 1
   lw_frame.inv_statusbar.of_BarIncrement(1)))
NEXT
```

of_SetBarStep

Description

Specifies the step value that the progress bar uses during a default increment process.

Access

Public

Syntax

instancename.**of_SetBarStep** (*stepvalue*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar
stepvalue	Integer specifying the default increment value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

The progress bar uses this value as the default when you call of_BarIncrement with no arguments.

Examples

This example calls the of_SetBarStep function:

```
Integer  li_return
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBarStep(5)
```

of_SetBarTextColor

Description

Specifies the progress bar's color.

Access

Public

Syntax

instancename.**of_SetBarTextColor** (*color*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
color	Long specifying progress bar color

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarTextColor function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBarTextColor &
    (RGB(255, 0, 0))
```

of_SetBarWidth

Description

Specifies a new width for the progress bar.

Access

Public

Syntax

instancename.**of_SetBarWidth** (*width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
width	Integer specifying a new progress bar width, in PBUs

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBarWidth function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBarWidth(300)
```

of_SetBorderType

Description

Specifies the default border type for new objects on the status bar.

Access

Public

Syntax

instancename.**of_SetBorderType** (*bordertype*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)

Argument	Description
bordertype	Integer specifying the border type. Values are: ♦ 0 None ♦ 1 Shadow ♦ 2 Rectangle ♦ 3 Resize ♦ 4 Line ♦ 5 3D lowered ♦ 6 3D raised

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetBorderType function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetBorderType(6)
lw_frame.inv_statusbar.of_Register &
    ("calc", "text", "Calculator", 300)
```

of_SetGapWidth

Description

Specifies the default between-object spacing for new objects on the status bar.

Access

Public

Syntax

instancename.**of_SetGapWidth** (*width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
width	Integer specifying the default between-object spacing for new status bar objects, in PBUs

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetGapWidth function:

```
...
this.inv_statusbar.of_SetGapWidth(20)
...
```

of_SetGDI

Description

Controls the display of GDI memory in the status bar.

Access

Public

Syntax

instancename.**of_SetGDI** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
boolean	Boolean specifying whether to display GDI memory in the status bar (TRUE) or remove GDI memory from the status bar (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

GDI display is not available in 32-bit Windows.

Examples

This example calls the of_SetGDI function:

```
w_frame.inv_statusbar.of_SetGDI(TRUE)
```

of_SetGDIThreshold

Description

Establishes a new GDI memory threshold value. The w_statusbar window displays GDI memory in a different color when it goes below the threshold value.

Access

Public

Syntax

instancename.**of_SetGDIThreshold** (*threshold*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
threshold	Long specifying the GDI memory threshold value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

GDI display is not available in 32-bit Windows.

Examples

This example calls the of_SetGDIThreshold function:

```
Integer   li_threshold
```

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
li_threshold = &
    lw_frame.inv_statusbar.of_GetGDIThreshold()
MessageBox("Status Bar", &
    "GDI memory threshold is " &
       + String(li_threshold))
```

of_SetGDIWidth

Description — Specifies the width for GDI display.

Access — Public

Syntax — *instancename*.**of_SetGDIWidth** (*width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
width	Integer specifying the width, in PBUs, for GDI memory display

Return value — Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage — GDI display is not available in 32-bit Windows.

Examples — This example calls the of_SetGDIWidth function:

```
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetGDIWidth(300)
```

of_SetMem

Description — Controls the display of free memory in the status bar.

Access — Public

Syntax — *instancename*.**of_SetMem** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
boolean	Boolean specifying whether to display free memory in the status bar (TRUE) or remove free memory from the status bar (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMem function:

```
w_frame.inv_statusbar.of_SetMem(TRUE)
```

of_SetMemThreshold

Description

Establishes a new free memory threshold value. The w_statusbar window displays free memory in a different color when it goes below the threshold value.

Access

Public

Syntax

instancename.**of_SetMemThreshold** (*threshold*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
threshold	Long specifying the free memory threshold value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMemThreshold function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetMemThreshold &
   (12000000)
```

of_SetMemWidth

Description

Specifies the width for free memory display.

Access

Public

Syntax

instancename.**of_SetMemWidth** (*width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
width	Integer specifying the width, in PBUs, for free memory display

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetMemWidth function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetMemWidth(300)
```

of_SetRefreshRate

Description

Sets a new refresh rate for free memory, GDI memory, and user memory.

Access

Public

Syntax

instancename.**of_SetRefreshRate** (*refreshinterval*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
refreshinterval	Integer specifying the number of seconds between refreshes

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetRefreshRate function:

```
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetRefreshRate(30)
```

of_SetTimer

Description

Enables the display of the date and time in the status bar.

Access

Public

Syntax

instancename.**of_SetTimer** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
boolean	Boolean specifying whether to display the date and time in the status bar (TRUE) or remove date and time display from the status bar (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

You customize date and time display format by calling the of_SetTimerFormat function.

Examples

This example calls the of_SetTimer function:

```
w_frame.inv_statusbar.of_SetTimer(TRUE)
```

of_SetTimerFormat

Description

Specifies a new date and time format for the timer object.

Access

Public

Syntax

instancename.**of_SetTimerFormat** (*format*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
format	String specifying the new date and time format. Sample formats include *mm/dd/yy h:mm:ss*, and *mmm dd, yyyy hh:mm:ss AM/PM*

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetTimerFormat function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetTimerInterval(60000)
lw_frame.inv_statusbar.of_SetTimerWidth(650)
lw_frame.inv_statusbar.of_SetTimerFormat &
   ("mmm dd, yyyy hh:mm AM/PM")
```

of_SetTimerInterval

Description

Specifies a new refresh rate for the timer object.

Access

Public

Syntax

instancename.**of_SetTimerInterval** (*interval*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
interval	Long specifying the number of milliseconds between timer refreshes (the default refresh rate is 1000 milliseconds)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

This function applies to the timer object only. To update the refresh rate for free memory, GDI memory, and user memory, call the of_SetRefreshRate function.

Examples

This example calls the of_SetTimerInterval function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetTimerInterval(60000)
lw_frame.inv_statusbar.of_SetTimerWidth(650)
lw_frame.inv_statusbar.of_SetTimerFormat &
    ("mmm dd, yyyy hh:mm AM/PM")
```

of_SetTimerWidth

Description

Specifies the width for the timer.

Access

Public

Syntax

instancename.**of_SetTimerWidth** (*width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
width	Integer specifying the width, in PBUs, for timer display

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetTimerWidth function:

```
w_frame   lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetTimerInterval(60000)
lw_frame.inv_statusbar.of_SetTimerWidth(650)
lw_frame.inv_statusbar.of_SetTimerFormat &
    ("mmm dd, yyyy hh:mm AM/PM")
```

of_SetUser

Description

Enables the display of user memory in the status bar.

Access

Public

Syntax

instancename.**of_SetUser** (*boolean*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
boolean	Boolean specifying whether to display user memory in the status bar (TRUE) or remove user memory from the status bar (FALSE)

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

User memory display is not available in 32-bit Windows.

Examples

This example calls the of_SetUser function:

```
w_frame.inv_statusbar.of_SetUser(TRUE)
```

of_SetUserThreshold

Description

Establishes a new user memory threshold value. The w_statusbar window displays user memory in a different color when it goes below the threshold value.

Access

Public

Syntax

instancename.**of_SetUserThreshold** (*threshold*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
threshold	Long specifying the user memory threshold value

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

GDI display is not available in 32-bit Windows.

Examples

This example calls the of_SetUserThreshold function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetUserThreshold &
    (12000000)
```

of_SetUserWidth

Description

Specifies the width for user memory display.

Access

Public

Syntax

instancename.**of_SetUserWidth** (*width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
width	Integer specifying the width, in PBUs, for user memory display

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_SetUserWidth function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_SetUserWidth(300)
```

of_UnRegister

Description

Removes a user-defined status bar object from the w_statusbar display.

Access

Public

Syntax

instancename.**of_UnRegister** (*id*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	String specifying the ID of the user-defined object to remove from the status bar. This is the ID specified in the of_Register function

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

To remove predefined objects, call the appropriate of_Set function (of_SetGDI, of_SetMem, and so on) passing an argument of FALSE.

Examples

This example calls the of_UnRegister function:

```
w_frame  lw_frame

lw_frame = gnv_app.of_GetFrame()
lw_frame.inv_statusbar.of_UnRegister("calc")
```

of_UnRegisterPredefined

Description

Unregisters a predefined status bar object.

Access

Protected

Syntax

instancename.**of_UnRegisterPredefined** (*id*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	String specifying one of the following. Values are: ♦ **Pfc_timer** Timer ♦ **Pfc_gdi** GDI memory ♦ **Pfc_user** User memory ♦ **Pfc_mem** Free memory ♦ **Pfc_progress** Progress bar

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SetMem function:

```
...
IF ib_mem THEN
   li_rc = of_RegisterPreDefined &
      ('pfc_mem', ii_memwidth)
ELSE
   li_rc = of_UnRegisterPreDefined('pfc_mem')
END IF
Return li_rc
```

of_UpdatePredefined

Description

Updates a registered predefined status bar object.

Access

Public

Syntax

instancename.**of_UpdatePredefined** (*id, width*)

Argument	Description
instancename	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
id	String specifying one of the following. Values are: ♦ **Pfc_timer** Timer ♦ **Pfc_gdi** GDI memory ♦ **Pfc_user** User memory ♦ **Pfc_mem** Free memory ♦ **Pfc_progress** Progress bar
width	Integer specifying a new width for the object

Return value

Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Usage

Internal.

Examples

This example is from the of_SetUserWidth function:

```
IF IsNull(ai_width) Or ai_width < 0 THEN
   Return -1
END IF
ii_userwidth = ai_width
```

```
of_UpdatePreDefined('pfc_user', ai_width)
Return 1
```

Index

A

Activate event
  w_frame 14
  w_master 24
  w_sheet 63
application INI file
  of_GetAppIniFile 588
  of_SetAppIniFile 595, 620
application manager 575
application services
  n_cst_apppreference 607
  n_cst_dwcache 679
  n_cst_error 989
  n_cst_security 1294
  n_cst_trregistration 1334

B

BeginDrag event 335
binary number service 1223
ButtonClicked event
  u_calculator 394
  u_calendar 421

C

calculator 391
calendar 416
CbnEditChange event
  u_cbx 134
  u_ddplb 141
CheckBox 129
Clicked event
  u_calendar 421
  u_dw 152
Close event
  w_frame 14
  w_master 24
CloseQuery event 24
code examples 10
color service 627
ColumnClick event 222
CommandButton 126
Commit 556
Connect 557
Constructor event
  n_cst_appmanager 580
  n_cst_debug 663
  n_cst_dwsrv_report 872
  n_cst_filesrvmac 1052
  n_cst_filesrvsol2 1073
  n_cst_filesrvwin16 1095
  n_cst_filesrvwin32 1112
  n_cst_inifile 1134
  n_cst_luw 1141
  n_cst_lvsrv_datasource 1166
  n_cst_platformmac 1240
  n_cst_platformsol2 1248
  n_cst_platformwin16 1257
  n_cst_platformwin32 1267
  n_cst_tvsrv_levelsource 1353
  n_cst_winsrv_sheetmanager 1428
  n_cst_winsrv_statusbar 1440
  u_calculator 394
  u_calendar 421
  u_progressbar 460
  u_st_splitbar 488
ContextInformation object 508
ContextKeyword object 509
conversion service 629

D

data type conversion service 629
DataStore 511

DataStore services
n_cst_dssrv 675
n_cst_dssrv_multitable 676
n_cst_dssrv_printpreview 677
n_cst_dssrv_report 678
DataWindow caching service
main discussion 679
of_SetDwCache 598
DataWindow control 146
DataWindow properties
n_cst_dwsrv_property 854
of_SetDWProperty 667
DataWindow services
base object 695
dropdown DataWindow search service 723
filter service 735
find service 747
linkage service 764
multitable update service 832
n_cst_dwsrv_printpreview 844
query mode service 858
reporting 870
required column 917
resize service 925
row manager 943
row selection 954
sort 972
u_dw 146
date/time service 642
DBError event
n_ds 514
u_dw 152
debug service
main discussion 661
of_SetDebug 598
Destructor event
n_cst_appmanager 580
n_cst_debug 663
n_cst_dwcache 681
n_cst_dwsrv_report 872
n_cst_error 993
n_cst_luw 1166
n_cst_trregistration 1335
n_cst_tvsrv_levelsource 1354
n_ds 514
n_tr 555
u_base 376
u_calculator 394
u_calendar 421
u_dw 152
u_em 199
u_lvs 222
u_rte 282
u_tvs 335
Disconnect 558
distributed computing
pfc_ConnectionBegin 581
pfc_ConnectionEnd 582
DoubleClicked event 422
DragDrop event 335
dropdown calculator 391
dropdown calendar 416
dropdown DataWindows
of_PopulateDDDW 716
of_PopulateDDDWs 717
of_RefreshDDDWs 717
search service 723
Dropdown event 152
dropdown service 669
DropDownListBox 132
DropDownPictureListBox 139
DynamicDescriptionArea object 510
DynamicStagingArea object 535

E

EditMask 198
EndLabelEdit event
u_lvs 222
u_tvs 335
Error object 536
error service
main discussion 989
of_SetError 599
examples, code 10

F

f_SetFilesrv 88
f_SetPlatform 89

file service
f_SetFilesrv 88
n_cst_filesrv 1019
n_cst_filesrvaix 1071
n_cst_filesrvhpux 1071
n_cst_filesrvmac 1050
n_cst_filesrvsol2 1071
n_cst_filesrvwin16 1093
FileExists event 282
filter service 735
find service 747
frame menu 71

G

GetFocus event
u_calculator 395
u_calendar 422
u_cb 126
u_cbx 129
u_ddlb 134
u_ddplb 141
u_dw 152
u_em 199
u_gr 207
u_hsb 210
u_lb 213
u_lvs 222
u_mlc 252
u_oc 258
u_p 265
u_pb 268
u_plb 271
u_rb 276
u_rte 282
u_sle 307
u_st 312
u_tv 335
u_vsb 369
Graph 207
GroupBox 206

H

Help
accessing 9
of_GetHelpFile 590
of_SetHelpFile 600
Hide event 15
HorizontalScrollBar 210

I

Inet object 538
INI file
of_GetAppIniFile 588
of_SetAppIniFile 595, 620
of_SetUserIniFile 603, 624
INI file service 1133
InternetResult object 539, 550
ItemChanged event 153
ItemError event 153
ItemExpanding event 336
ItemFocusChanged event 153

K

Key event
u_calculator 395
u_calendar 422
KeyDown event
u_lvs 222
u_tvs 336

L

LButtonDown event 153
LButtonUp event
u_dw 154
u_st_splitbar 488
linkage service 764
ListBox 213
ListView 218
ListView service
base 1157
data source 1164

sort 1203
logical unit of work service 1139
logon
of_LogonDlg 595
pfc_Logon 583
pfc_PreLogonDlg 585
LoseFocus event
u_calculator 395
u_calendar 422

M

m_dw 66
m_edit 69
m_frame 71
m_lvs 73
m_master 76
m_oc 82
m_tvs 84
MailSession object 540
master menu 76
master/detail processing 764
menu service 1210
Message object 542
metaclass service 1216
MicroHelp
of_GetMicrohelp 591
of_SetMicrohelp 601
pfc_MicroHelp 15, 28, 63
MouseMove event 489
Move event 15, 24
MultiLineEdit 251
multitable update service
n_cst_dssrv_multitable 676
n_cst_dssrv_printpreview 677
n_cst_dwsrv_multitable 832

N

n_base 572
n_cn 504
n_cst_aboutattrib 92
n_cst_appmanager 575
n_cst_apppreference 607
n_cst_baseattrib 93
n_cst_calculatorattrib 94
n_cst_calendarattrib 95
n_cst_color 627
n_cst_columnattrib 96
n_cst_conversion 629
n_cst_datetime 642
n_cst_dberrorattrib 97
n_cst_debug
main discussion 661
n_cst_sqlspy 1306
n_cst_dirattrib 98
n_cst_dropdown 669
n_cst_dssrv 675
n_cst_dssrv_multitable 676
n_cst_dssrv_printpreview 677
n_cst_dssrv_report 678
n_cst_dwcache 679
n_cst_dwobjectattrib 99
n_cst_dwpropertyattrib 100
n_cst_dwsrv 695
n_cst_dwsrv_dropdownsearch 723
n_cst_dwsrv_filter 735
n_cst_dwsrv_find 747
n_cst_dwsrv_linkage 764
n_cst_dwsrv_multitable 832
n_cst_dwsrv_printpreview 844
n_cst_dwsrv_property 854
n_cst_dwsrv_querymode 858
n_cst_dwsrv_report 870
n_cst_dwsrv_reqcolumn 917
n_cst_dwsrv_resize 925
n_cst_dwsrv_rowmanager 943
n_cst_dwsrv_rowselection 954
n_cst_dwsrv_sort 972
n_cst_error 989
n_cst_errorattrib 101
n_cst_filesrv 1019
n_cst_filesrvaix 1071
n_cst_filesrvhpux 1071
n_cst_filesrvmac 1050
n_cst_filesrvsol2 1071
n_cst_filesrvwin16 1093
n_cst_filesrvwin32 1110
n_cst_filterattrib 102
n_cst_findattrib 103

n_cst_infoattrib 104
n_cst_inifile 1133
n_cst_itemattrib 105
n_cst_linkageattrib 106
n_cst_logonattrib 107
n_cst_luw 1139
n_cst_lvsrv 1157
n_cst_lvsrv_datasource 1164
n_cst_lvsrv_sort 1203
n_cst_lvsrvattrib 108
n_cst_menu 1210
n_cst_metaclass 1216
n_cst_numerical 1223
n_cst_platform 1230
n_cst_platformaix 1246
n_cst_platformhpux 1246
n_cst_platformmac 1239
n_cst_platformsol2 1246
n_cst_platformwin16 1256
n_cst_platformwin32 1266
n_cst_resize 1276
n_cst_restorerowattrib 110
n_cst_returnattrib 111
n_cst_rtefind 1287
n_cst_security 1294
n_cst_selection 1298
n_cst_selectionattrib 112
n_cst_sortattrib 113
n_cst_splashattrib 114
n_cst_sql 1303
n_cst_sqlattrib 115
n_cst_sqlspy 1306
n_cst_string 1312
n_cst_textstyleattrib 116
n_cst_tmgregisterattrib 117
n_cst_toolbarattrib 118
n_cst_trregistration 1334
n_cst_tvsrv 1341
n_cst_tvsrv_levelsource 1351
n_cst_tvsrv_print 1401
n_cst_tvsrvattrib 119
n_cst_winsrv 1405
n_cst_winsrv_preference 1408
n_cst_winsrv_sheetmanager 1427
n_cst_winsrv_statusbar 1435
n_cst_zoomattrib 120
n_cxinfo 508
n_cxk 509
n_dda 510
n_ds 511
n_dsa 535
n_err 536
n_inet 538
n_ir 539
n_ms 540
n_msg 542
n_oo 544
n_ostg 545
n_ostm 546
n_pl 547
n_srv 549
n_tmg 550
n_tr
  main discussion 553
  n_cst_trregistration 1334
n_trp 569
naming conventions 2
numerical service 1223

O

of_About 588
of_AcceptText
  n_cst_dwsrv_linkage 778
  n_cst_luw 1141
  n_ds 520
  u_base 379
  u_dw 174
  u_lvs 234
  u_tab 319
  u_tvs 349
  w_master 42
of_AddColumn 726
of_AddCompute 873
of_AddItem
  u_lvs 235
  u_tvs 349
of_AddLine 876
of_AddPicture 880
of_AddText 881
of_AddToUpdate 834

of_ArrayToString 1313
of_Assemble 1303
of_AssemblePath 1021
of_BarIncrement 1445
of_BarReset 1446
of_BarUpdateVisuals 1446
of_Begin 556
of_BeginTran 1142
of_Binary 1224
of_BitwiseAnd 1224
of_BitwiseNot 1225
of_BitwiseOr 1226
of_BitwiseXor 1227
of_Boolean 630
of_BuildColumnNames 751
of_BuildComparison 697
of_BuildExpression 698
of_BuildFilterAttrib 737
of_BuildFindExpression 752
of_BuildSortAttrib 976
of_BuildTextModify 884
of_BuildTree 1402
of_Button 631
of_ButtonClicked 396
of_ButtonUp 961
of_CalculateFileAttributes 1022
of_CalculateMicrohelpHeight 1447
of_CanUndo
  n_cst_lvsrv_datasource 1171
  n_cst_tvsrv_levelsource 1358
  u_tvs 350
of_Center 1406
of_ChangeDirectory
  n_cst_filesrv 1023
  n_cst_filesrvmac 1052
  n_cst_filesrvsol2 1074
  n_cst_filesrvwin16 1095
  n_cst_filesrvwin32 1113
of_CheckRequired
  n_cst_dwsrv_linkage 778
  n_ds 521
  u_dw 174
of_ClearLog 663
of_ClearUndo
  n_cst_lvsrv_datasource 1172
  n_cst_tvsrv_levelsource 1359
of_Commit 556
of_CompareAttrib
  n_cst_dwsrv_find 752
  n_cst_rtefind 1290
of_ConfirmDelete
  n_cst_dwsrv_rowmanager 946
  n_cst_lvsrv_datasource 1172
  n_cst_tvsrv_levelsource 1360
of_ConfirmOnDelete 779
of_ConfirmOnRowChange 780
of_Connect 557
of_ConvertDate 1096
of_ConvertFileDateTimeToPB
  n_cst_filesrvmac 1053
  n_cst_filesrvsol2 1075
  n_cst_filesrvwin32 1114
of_ConvertPBDateTimeToFile
  n_cst_filesrvmac 1054
  n_cst_filesrvsol2 1076
  n_cst_filesrvwin32 1114
of_ConvertTime 1097
of_ConvertToRow 1173, 1360
of_CopyTo 558
of_CountOccurrences 1313
of_CreateComposite 887
of_CreateDirectory
  n_cst_filesrv 1023
  n_cst_filesrvmac 1055
  n_cst_filesrvsol2 1077
  n_cst_filesrvwin16 1097
  n_cst_filesrvwin32 1115
of_CreateDWObject 1447
of_CreateLogText 994
of_CreateNotifyText 994
of_CreateUpdateSequence 781
of_Date 631
of_DayOfWeek 643
of_Days 644
of_DBError 1143
of_Decimal 1228
of_Delete 1134
of_DeleteAll 946
of_DeleteDetailRows 782
of_DeleteItem 727
of_DeleteRow 947
of_DeleteRows 782

of_DeleteSelected 948
of_DelTree 1024
of_Describe 699
of_DirAttribToDS 1024
of_DirectoryExists
n_cst_filesrv 1025
n_cst_filesrvmac 1055
n_cst_filesrvsol2 1077
n_cst_filesrvwin16 1098
n_cst_filesrvwin32 1116
of_DirList
n_cst_filesrv 1026
n_cst_filesrvmac 1056
n_cst_filesrvwin16 1099
n_cst_filesrvwin32 1117
of_Disconnect 558
of_DistinctValues 559
of_DrawMonth 425
of_DropDown
u_calculator 397
u_calendar 426
of_DSToDirAttrib 1026
of_DWArguments 701
of_DWItemStatus 632
of_End 559
of_EndTran 1144
of_EndTranError 1144
of_Execute 560
of_FileCopy 1027
of_FileRead 1028
of_FileRename
n_cst_filesrv 1030
n_cst_filesrvmac 1058
n_cst_filesrvsol2 1078
n_cst_filesrvwin16 1100
n_cst_filesrvwin32 1118
of_FileWrite 1031
of_Filter 738
of_FilterDetails 783
of_Find 753
of_FindFirstItemLevel 1342
of_FindItem 1343
of_FindMatchingEvent 1217
of_FindMatchingVariable 1218
of_FindNext 1158
of_FindRoot 784
of_FindSelected 1159
of_FindStartAndEndRows 754
of_FindWindow
n_cst_platform 1231
n_cst_platformsol2 1248
n_cst_platformwin16 1258
n_cst_platformwin32 1268
of_FirstDayOfMonth 644
of_FirstPage 845
of_FormatData 1173
of_GetAllToolbarIndex 1211
of_GetAltFilename
n_cst_filesrv 1032
n_cst_filesrvsol2 1079
n_cst_filesrvwin32 1119
of_GetAlwaysOnTop
n_cst_debug 664
n_cst_sqlspy 1307
of_GetAncestorClasses 1219
of_GetAppend 522
of_GetAppIniFile 588
of_GetAppKey 589
of_GetArgs 1361
of_GetArguments 785
of_GetAutoRollback 560
of_GetBackColor 461
of_GetBar 1448
of_GetBarAutoClear 1462
of_GetBarColor 489
of_GetBarDisplayStyle 1449
of_GetBarFillColor 1449
of_GetBarFillStyle 1450
of_GetBarMaximum 1451
of_GetBarMinimum 1451
of_GetBarMoveColor 490
of_GetBarPctComplete 1452
of_GetBarPosition 1452
of_GetBarStep 1453
of_GetBarTextColor 1454
of_GetBarWidth 1454
of_GetBeep 995
of_GetBit 1228
of_GetBorderType 1455
of_GetByName 1336
of_GetCloseStatus 42
of_GetColumn 727

of_GetColumnDisplayName 703
of_GetColumnDisplayNameStyle 704
of_GetColumnEditStatus 755
of_GetColumnHeader 976
of_GetColumnInfo 1174
of_GetColumnLabel 1177, 1366
of_GetColumnNameSource 705
of_GetComputerName
  n_cst_platform 1232
  n_cst_platformmac 1241
  n_cst_platformsol2 1249
  n_cst_platformwin32 1269
of_GetConfirmOnDelete 948
of_GetContinuousPages 290
of_GetCopyright 589
of_GetCount
  n_cst_dwcache 682
  n_cst_trregistration 1337
of_GetCreationDate 1033
of_GetCreationDateTime
  n_cst_filesrv 1033
  n_cst_filesrvmac 1059
  n_cst_filesrvsol2 1079
  n_cst_filesrvwin32 1120
of_GetCreationTime 1034
of_GetCurrentDirectory
  n_cst_filesrv 1035
  n_cst_filesrvmac 1060
  n_cst_filesrvsol2 1080
  n_cst_filesrvwin16 1101
  n_cst_filesrvwin32 1121
of_GetCurrentState 1430
of_GetCurrentText 461
of_GetCustomUpdate 785
of_GetDataAny 1344
of_GetDataNumeric 1345
of_GetDataObject
  n_cst_lvsrv_datasource 1175
  n_cst_tvsrv_levelwsource 1362
of_GetDataRow 1175, 1362
of_GetDataSource 1176
of_GetDataStore 1363
of_GetDataString 1345
of_GetDefaultBackColor 889
of_GetDefaultBorder 889
of_GetDefaultCharset 890
of_GetDefaultColor 890
of_GetDefaultFontFace 891
of_GetDefaultFontSize 892
of_GetDefaultHeaderSuffix 705
of_GetDeleteStyle 786
of_GetDetails 786
of_GetDiskSpace
  n_cst_filesrv 1035
  n_cst_filesrvmac 1060
  n_cst_filesrvsol2 1081
  n_cst_filesrvwin16 1101
  n_cst_filesrvwin32 1121
of_GetDisplayItem
  n_cst_dwsrv 706
  n_cst_lvsrv 1160
  n_cst_tvsrv 1346
of_GetDisplayStyle 462
of_GetDisplayUnits
  n_cst_dwsrv 706
  n_cst_lvsrv 1160
  n_cst_tvsrv 1346
of_GetDriveType
  n_cst_filesrv 1036
  n_cst_filesrvmac 1061
  n_cst_filesrvsol2 1082
  n_cst_filesrvwin16 1102
  n_cst_filesrvwin32 1122
of_GetDWType 43
of_GetEnabled
  n_cst_dwsrv_printpreview 846
  n_cst_dwsrv_querymode 860
of_GetExclude
  n_cst_dwsrv_filter 739
  n_cst_dwsrv_sort 977
  n_cst_lvsrv_sort 1206
of_GetExtremePoint 490
of_GetFileAttributes
  n_cst_filesrv 1036
  n_cst_filesrvmac 1062
  n_cst_filesrvsol2 1083
  n_cst_filesrvwin16 1103
  n_cst_filesrvwin32 1123
of_GetFileName 291
of_GetFileSize
  n_cst_filesrv 1037
  n_cst_filesrvmac 1063

n_cst_filesrvsol2 1084
n_cst_filesrvwin16 1104
n_cst_filesrvwin32 1124
of_GetFillColor 463
of_GetFillStyle 463
of_GetFilter 739
of_GetFrame 590
of_GetFreeMemory
n_cst_platform 1232
n_cst_platformmac 1241
n_cst_platformsol2 1249
n_cst_platformwin16 1258
n_cst_platformwin32 1269
of_GetFreeResources
n_cst_platform 1233
n_cst_platformwin16 1259
of_GetGapWidth 1456
of_GetGDI 1456
of_GetGDIThreshold 1457
of_GetHeaderName 707
of_GetHeight 708
of_GetHelpFile 590
of_GetHoliday 426
of_GetHolidayColor 427
of_GetHorizontalPointer 491
of_GetIgnoreFileExists 291
of_GetInfo
n_cst_dwsrv 708
n_cst_dwsrv_dropdownsearch 728
n_cst_dwsrv_filter 740, 756
n_cst_dwsrv_linkage 787
n_cst_dwsrv_multitable 835
n_cst_dwsrv_printpreview 846
n_cst_dwsrv_querymode 861
n_cst_dwsrv_report 892
n_cst_dwsrv_reqcolumn 918
n_cst_dwsrv_resize 929
n_cst_dwsrv_rowmanager 949
n_cst_dwsrv_rowselection 961
n_cst_dwsrv_sort 977
u_base 380
u_calculator 397
u_calendar 427
u_lvs 235
u_progressbar 464
u_st_splitbar 491
u_tab 320
u_tvs 351
w_master 44
of_GetItem 709
of_GetItemAny 710
of_GetItemForData 1364
of_GetItemHandle 1365
of_GetItemIndex 1177
of_GetKeyboard 964
of_GetKeys 1135
of_GetKeyValue 1314
of_GetLastAccessDate
n_cst_filesrv 1038
n_cst_filesrvmac 1064
n_cst_filesrvsol2 1085
n_cst_filesrvwin32 1125
of_GetLastWriteDate 1038
of_GetLastWriteDateTime
n_cst_filesrv 1039
n_cst_filesrvsol2 1086
n_cst_filesrvwin16 1105
n_cst_filesrvwin32 1126
of_GetLastWriteTime 1039
of_GetLevel 1366
of_GetLevelAttributes 1367
of_GetLevelCount 1368
of_GetLineEnding 1136
of_GetLogFile 996
of_GetLogFileStyle 996
of_GetLogo 591
of_GetLogSeverity 997
of_GetLongFilename
n_cst_filesrv 1040
n_cst_filesrvsol2 1087
n_cst_filesrvwin32 1127
of_GetMarkedDay 428
of_GetMarkedDayColor 429
of_GetMaster 788
of_GetMaximum 464
of_GetMDIFrame 1211
of_GetMem 1458
of_GetMemThreshold 1458
of_GetMenuItems 1411
of_GetMenuReference 1212
of_GetMessageText 465
of_GetMethod

n_cst_lvsrv_datasource 1178
n_cst_tvsrv_levelsource 1368
of_GetMicrohelp 591
of_GetMinimum 465
of_GetMinMaxPoints 1280
of_GetMinObjectSize 492
of_GetName 561
of_GetNextLevel 352
of_GetNotifyConnection 997
of_GetNotifyPreTitle 998
of_GetNotifySeverity 999
of_GetNotifyWho 999
of_GetObjectInformation 930
of_GetObjects
n_cst_dwsrv 711
n_cst_tvsrv_levelsource 1369
u_lvs 236
u_tvs 352
of_GetOtherSaveObjects 788
of_GetPageInputField 292
of_GetParentPosition 670
of_GetParentWindow
n_cst_dropdown 671
n_ds 522, 527
u_base 380
u_cb 127
u_cbx 130
u_ddlb 136
u_ddplb 143
u_dw 175
u_em 203
u_gr 208
u_hsb 211
u_lb 214
u_lvs 237
u_mle 255
u_oc 262
u_p 266
u_pb 269
u_plb 272
u_rb 277
u_rte 292
u_sle 310
u_st 313
u_tab 321
u_tvs 353
u_vsb 370
of_GetPctComplete 466
of_GetPhysicalMemory
n_cst_platform 1233
n_cst_platformmac 1242
n_cst_platformsol2 1250
n_cst_platformwin32 1270
of_GetPictureColumn
n_cst_lvsrv_datasource 1179
n_cst_tvsrv_levelsource 1371
of_GetPictureSize 893
of_GetPosition 467
of_GetPredefinedSource 1000
of_GetPredefinedSourceType 1001
of_GetPropertyInfo
n_cst_dwsrv 713
n_cst_dwsrv_dropdownsearch 729
n_cst_dwsrv_filter 741, 757
n_cst_dwsrv_linkage 789
n_cst_dwsrv_multitable 835
n_cst_dwsrv_printpreview 847
n_cst_dwsrv_querymode 862
n_cst_dwsrv_report 894
n_cst_dwsrv_reqcolumn 919
n_cst_dwsrv_resize 931
n_cst_dwsrv_rowmanager 950
n_cst_dwsrv_rowselection 962
n_cst_dwsrv_sort 978
u_calculator 398
u_calendar 429
of_GetQueryCols 862
of_GetRefreshRate 1459
of_GetRegisterable
n_cst_dwsrv_dropdownsearch 729
n_cst_dwsrv_filter 741
n_cst_dwsrv_multitable 836
n_cst_dwsrv_reqcolumn 920
n_cst_dwsrv_resize 932
n_cst_dwsrv_sort 979
u_calculator 399
u_calendar 430
of_GetRegisterableColumn 837
of_GetRegisterableTable 838
of_GetRegistered
n_cst_dwcache 682
n_cst_dwsrv_dropdownsearch 730

n_cst_dwsrv_linkage 790
n_cst_dwsrv_multitable 838
n_cst_dwsrv_reqcolumn 920
n_cst_dwsrv_resize 933
n_cst_trregistration 1337
u_calculator 399
u_calendar 431
of_GetRegisteredStyle
u_calculator 400
u_calendar 432
of_GetResetCriteria 863
of_GetRestoreRow 951
of_GetRetrieveArgs 1371
of_GetRetrieveOnDisabled 864
of_GetRow 237
of_GetRuler 848
of_GetSaturdayColor 432
of_GetSaveSound 791
of_GetSaveStatus 44
of_GetSections 1136
of_GetSelected
u_lb 215
u_plb 273
of_GetSelectedPictureColumn 1372
of_GetSeparator 1041
of_GetSheetCount 1430
of_GetSheets 1431
of_GetSheetsByClass 1431
of_GetSheetsByTitle 1432
of_GetSort 980
of_GetSqlState 562
of_GetStartPageNumber 293
of_GetStatePictureColumn
n_cst_lvsrv_datasource 1180
n_cst_tvsrv_levelsource 1373
of_GetStep 467
of_GetStyle
n_cst_dwsrv_filter 742
n_cst_dwsrv_linkage 791
n_cst_dwsrv_rowselection 963
n_cst_dwsrv_sort 980
n_cst_error 1001
of_GetSundayColor 433
of_GetSystemDirectory
n_cst_platform 1234
n_cst_platformmac 1242
n_cst_platformsol2 1250
n_cst_platformwin16 1259
n_cst_platformwin32 1270
of_GetSystemSetting 672
of_GetTextColor 468
of_GetTextSize
n_cst_platform 1234
n_cst_platformsol2 1251
n_cst_platformwin16 1260
n_cst_platformwin32 1271
of_GetTextSizePos 894
of_GetTextStyle 293
of_GetTimeOut 1002
of_GetTimer 1459
of_GetTimerInterval 1460
of_GetToken 1315
of_GetToolbarItemOrder 1411
of_GetToolbarItemSpace 1412
of_GetToolbarItemVisible 1412
of_GetToolbars 1413
of_GetToolbarTitles 1414
of_GetTrace 562
of_GetTransObject
n_cst_lvsrv_datasource 1180
n_cst_tvsrv_levelsource 1373
of_GetType 1145
of_GetTypeToProcess 1145
of_GetUnattended 1003
of_GetUndoLevels 896
of_GetUpdateable 176
of_GetUpdateBottomUp 792
of_GetUpdateObjects
u_base 381
u_tab 321
w_master 45
of_GetUpdateRequestor
n_cst_luw 1146
u_lvs 238
u_tvs 353
of_GetUpdatesPending 792
of_GetUpdateStyle 793
of_GetUseColLinks 794
of_GetUseDisplay 981
of_GetUser
n_cst_error 1003
n_cst_winsrv_statusbar 1461

of_GetUserID
n_cst_appmanager 592
n_cst_platform 1235
n_cst_platformmac 1243
n_cst_platformsol2 1252
n_cst_platformwin16 1261
n_cst_platformwin32 1272
of_GetUserIniFile 592
of_GetUserKey 593
of_GetUserThreshold 1461
of_GetValue 794
of_GetVersion 593
of_GetVerticalPointer 492
of_GetVisibleOnly
n_cst_dwsrv_filter 743
n_cst_dwsrv_sort 982
of_GetVolumes
n_cst_filesrv 1041
n_cst_filesrvmac 1065
of_GetWidth 713
of_GetWindow 1414
of_GetWindowsDirectory
n_cst_platform 1236
n_cst_platformsol2 1252
n_cst_platformwin16 1262
n_cst_platformwin32 1272
of_GetWindowText
n_cst_platform 1236
n_cst_platformsol2 1253
n_cst_platformwin16 1262
n_cst_platformwin32 1273
of_GetXPosColumn 1181
of_GetYPosColumn 1182
of_GetZoom 848
of_GlobalReplace 1316
of_Gregorian 645
of_Hours 646
of_Icon 633
of_IncludeFile
n_cst_filesrv 1042
n_cst_filesrvmac 1066
of_Increment 468
of_Init
n_cn 505
n_tr 563
of_Initialize
n_cst_dwsrv_find 757
n_cst_rtefind 1291
of_InitSecurity 1295
of_InsertDocument 295
of_InsertItem
n_cst_lvsrv_datasource 1182
n_cst_tvsrv_levelsource 1374
u_lvs 238, 239, 240
u_tvs 354, 355
of_InsertPicture 295
of_InsertRow 950
of_Integer 634
of_InvertSelection 964
of_IsAllowFindDlg 758
of_IsAllowReplaceDlg 759
of_IsAlpha 1317
of_IsAlphaNum 1317
of_IsAlwaysRedraw 433
of_IsAlwaysValidate
n_cst_luw 1147
u_base 382
u_lvs 240
u_tab 322
u_tvs 356
w_master 45
of_IsAncestorClass 1219
of_IsAppRunning
n_cst_platform 1237
n_cst_platformwin16 1263
of_IsArithmeticOperator 1318
of_IsAutoReset 469
of_IsCloseOnClick
u_calculator 401
u_calendar 434
of_IsCloseOnDClick 435
of_IsColumnHeader 1206
of_IsComparisonOperator 1319
of_IsConfirmOnDelete
n_cst_dwsrv_linkage 795
n_cst_lvsrv_datasource 1183
n_cst_tvsrv_levelsource 1375
of_IsConfirmOnRowChange 795
of_IsConnected 564
of_IsDateType 435
of_IsDropDown 401

of_IsEmpty 1319
of_IsEventDefined 1220
of_IsEventImplemented 1221
of_IsExclude 1207
of_IsFormat 1320
of_IsFunctionDefined 1222
of_IsHolidayBold 436
of_IsInArray 1213
of_IsInitialValue
  u_calculator 402
  u_calendar 437
of_IsKey 796
of_IsLeapYear 646
of_IsLinked 796
of_IsLogOpen 664
of_IsLower 1321
of_IsMarkedDayBold 437
of_IsNumericType 403
of_IsOngoingFind
  n_cst_dwsrv_find 759
  n_cst_rtefind 1292
of_IsOperator 404
of_IsPredefined 1463
of_IsPrintable 1321
of_IsPropertyOpen 855
of_IsPunctuation 1322
of_IsRecursiveLevel 1376
of_IsRegistered
  n_cst_dwcache 685
  n_cst_dwsrv_dropdownsearch 731
  n_cst_dwsrv_multitable 839
  n_cst_dwsrv_reqcolumn 921
  n_cst_dwsrv_resize 933
  n_cst_trregistration 1338
  u_calculator 404
  u_calendar 438
of_IsRegistryAvailable 594
of_IsRestoreApp 611
of_IsRestoreUser 611
of_IsRmbMenu 357
of_IsRoot
  n_cst_dwsrv_linkage 797
  u_dw 176
of_IsSaturdayBold 438
of_IsSelfUpdatingObject 1147
of_IsSharedProperty 177
of_IsSpace 1322
of_IsSundayBold 439
of_IsSyncOnKeyChange 798
of_IsUpdateable
  n_ds 523
  u_base 382
  u_dw 177
  u_lvs 241
  u_tab 323
  u_tvs 357
  w_master 46
of_IsUpdateOnRowChange 798
of_IsUpper 1323
of_IsValid 647
of_IsWeekDay 648
of_IsWeekEnd 648
of_IsWhiteSpace 1324
of_ItemFocusChanged 799
of_Julian 649
of_JulianDayNumber 650
of_KeyChanged 799
of_KeySync 800
of_LastDayOfMonth 650
of_LastPage 849
of_LastPos 1324
of_LButtonUp 493
of_LeftTrim 1325
of_LinkDetail 801
of_LinkTo 802
of_Load 864
of_LoadPredefinedMsg 1004
of_LogonDlg 595
of_Message
  n_cst_debug 665
  n_cst_error 1005
of_MessageBox
  n_base 572
  n_cn 506
  n_ds 523
  n_err 536
  n_ms 540
  n_msg 542
  n_pl 547
  n_tr 564
  n_trp 569
  u_base 383

u_cb 127
u_cbx 130
u_ddlb 137
u_ddplb 144
u_dw 178
u_em 203
u_gr 208
u_hsb 211
u_lb 216
u_lvs 241
u_mle 255
u_oc 263
u_p 266
u_pb 269
u_plb 274
u_rb 277
u_rte 296
u_sle 310
u_st 313
u_tab 323
u_tvs 358
u_vsb 370
w_master 46
of_MilliSecsAfter 651
of_Modify
n_cst_dwsrv 714
n_cst_winsrv_statusbar 1463
of_MonthsAfter 652
of_MouseMove 493
of_Open
n_cst_selection 1299
n_cst_winsrv_statusbar 1464
of_OpenDocument 297
of_OpenLog 665
of_OpenProperty 856
of_OpenPropertyService 856
of_OpenSQLSpy 1308
of_OSType 634
of_PadLeft 1326
of_PadRight 1327
of_Page 849
of_PageAcross 850
of_PageCount 850
of_PageCountAcross 851
of_PageSetupDlg 1237
of_Parse 1304
of_ParseArgs 1376
of_ParsePath 1042
of_ParseSortAttrib 982
of_ParseToArray 1327
of_PerformMath 405
of_PlaySound
n_cst_platform 1238
n_cst_platformmac 1244
n_cst_platformsol2 1254
n_cst_platformwin16 1263
n_cst_platformwin32 1273
of_Populate 359
of_PopulateDDDW 716
of_PopulateDDDWs 717
of_Position 673
of_PostUpdate
n_cst_dwsrv_linkage 802
n_cst_luw 1148
n_ds 525
u_base 384
u_dw 179
u_lvs 243
u_tab 324
u_tvs 359
w_master 48
of_PrepPrint 897
of_PreUpdate 1148
of_PrintDlg
n_cst_platform 1238
n_cst_platformmac 1244
n_cst_platformsol2 1254
n_cst_platformwin16 1264
n_cst_platformwin32 1274
of_PrintLog 666
of_PrintReport 898
of_PrintTree 1402
of_ProcessLog 1007
of_ProcessMessage 1008
of_ProcessMessageSubstitution 1008
of_ProcessNotify 1009
of_Quote 1328
of_RedirectFocus
u_calculator 406
u_calendar 440
of_Redraw 494

of_Refresh
n_cst_dwcache 685
n_cst_dwsrv_linkage 803
n_cst_lvsrv_datasource 1184
of_RefreshDDDWs 717
of_RefreshItem 1377
of_RefreshLevel 1378
of_Register
n_cst_dwcache 686, 688
n_cst_dwsrv_dropdownsearch 731
n_cst_dwsrv_linkage 803
n_cst_dwsrv_multitable 840
n_cst_dwsrv_resize 934
n_cst_lvsrv_datasource 1184
n_cst_resize 1281
n_cst_trregistration 1339
n_cst_tvsrv_levelsource 1379
n_cst_winsrv_statusbar 1465
u_calculator 406
u_calendar 440
u_st_splitbar 495
of_RegisterArgs 692
of_RegisterDataSource
n_cst_lvsrv_datasource 1191
n_cst_tvsrv_levelsource 1388
of_RegisterPredefined 1466
of_RegisterReportColumn 1192
of_RegisterSkipColumn 922
of_RelativeDateTime 653
of_RelativeMonth 654
of_RelativeYear 655
of_RemoveChildren 1389
of_RemoveColumn 732
of_RemoveDirectory
n_cst_filesrv 1043
n_cst_filesrvmac 1067
n_cst_filesrvsol2 1088
n_cst_filesrvwin16 1106
n_cst_filesrvwin32 1128
of_RemoveNonPrint 1329
of_RemoveWhiteSpace 1329
of_Replace 760
of_Reset
n_cst_dwsrv_linkage 804
n_cst_lvsrv_datasource 1193
n_cst_tvsrv_levelsource 1390
u_calculator 409
u_calendar 443
u_dw 180
u_lvs 243
u_progressbar 469
u_tvs 360
of_ResetArguments 804
of_ResetList 1194
of_ResetMaster 805
of_ResetTree 1390
of_ResetUndo 898
of_ResetUpdate
n_cst_dwsrv_linkage 805
n_cst_lvsrv_datasource 1194
n_cst_tvsrv_levelsource 1391
of_Resize
n_cst_dwsrv_resize 938
n_cst_resize 1283
of_Restore
n_cst_apppreference 612
n_cst_winsrv_preference 1415
of_RestoreApp 614
of_RestoreFocusPoint 1467
of_RestoreMenu 1417
of_RestorePositiveNumber 1417
of_RestoreUpdateSettings 841
of_RestoreUser 615
of_Retrieve
n_cst_dwsrv_linkage 806
n_cst_lvsrv_datasource 1195
n_cst_tvsrv_levelsource 1391
n_ds 525
u_dw 180
u_tvs 360
of_RetrieveDetails 807
of_RightTrim 1330
of_Rollback 565
of_RowSelect 966
of_RowSelectExt 967
of_RowSelectMulti 967
of_RowSelectSingle 968
of_Save
n_cst_apppreference 616
n_cst_dwsrv_linkage 808
n_cst_dwsrv_querymode 865
n_cst_luw 1149

n_cst_winsrv_preference 1418
of_SaveApp 618
of_SaveMenu 1420
of_SaveUser 619
of_ScrollDetails 809
of_SearchChild 1347
of_SearchItem 733
of_SecondsAfter 655
of_SelectedCount 969
of_SelectText 761
of_SendMessage
m_master 80
n_cst_menu 1213
of_SetAllowFindDlg 762
of_SetAllowReplaceDlg 762
of_SetAlwaysOnTop
n_cst_debug 666
n_cst_sqlspy 1308
of_SetAlwaysRedraw 443
of_SetAlwaysValidate
n_cst_luw 1150
u_base 384
u_lvs 244
u_tab 325
u_tvs 361
w_master 48
of_SetAppend 526
of_SetAppIniFile
n_cst_appmanager 595
n_cst_apppreference 620
of_SetAppKey 596, 621
of_SetAppPreference 597
of_SetArguments 810
of_SetAutoReset 470
of_SetAutoRollback 566
of_SetBackColor 470
of_SetBackground 899
of_SetBar 1467
of_SetBarAutoClear 1468
of_SetBarColor 496
of_SetBarFillColor 1469
of_SetBarFillStyle 1470
of_SetBarMax 1471
of_SetBarMoveColor 496
of_SetBarOffsetX 1472
of_SetBarOffsetY 1472
of_SetBarPosition 1473
of_SetBarStep 1474
of_SetBarStyle 1469
of_SetBarTextColor 1474
of_SetBarWidth 1475
of_SetBase
n_ds 526, 551
n_tmg 551
u_dw 181
u_lvs 244
u_tvs 362
w_master 49
of_SetBatchMode 1309
of_SetBeep 1010
of_SetBorder 902
of_SetBorderType 1475
of_SetCache
n_cst_lvsrv_datasource 1195
n_cst_tvsrv_levelsource 1392
of_SetCloseOnClick
u_calculator 409
u_calendar 444
of_SetCloseOnDClick 445
of_SetColor 903
of_SetColumnDisplayNameStyle 718
of_SetColumnHeader
n_cst_dwsrv_sort 983
n_cst_lvsrv_sort 1208
of_SetColumnNameSource 718
of_SetConfirmOnDelete
n_cst_dwsrv_linkage 810
n_cst_dwsrv_rowmanager 952
n_cst_lvsrv_datasource 1196
n_cst_tvsrv_levelsource 1393
of_SetConfirmOnRowChange 811
of_SetContinuousPages 298
of_SetCopyright 597
of_SetCreationDateTime
n_cst_filesrv 1044
n_cst_filesrvmac 1068
n_cst_filesrvsol2 1089
n_cst_filesrvwin32 1128
of_SetCurrentState 1433
of_SetCustomUpdate 811
of_SetDataSource
u_lvs 245

u_tvs 362
of_SetDate 445
of_SetDateFormat 446
of_SetDBErrorMsg
n_cst_luw 1151
w_master 49
of_SetDebug 598
of_SetDefaultBackColor 905
of_SetDefaultBorder 906
of_SetDefaultCharset 906
of_SetDefaultColor 907
of_SetDefaultFontFace 908
of_SetDefaultFontSize 908
of_SetDefaultHeaderSuffix 719
of_SetDeleteStyle
n_cst_dwsrv_linkage 812
n_cst_tvsrv_levelsource 1393
of_SetDisplayItem
n_cst_dwsrv 719
n_cst_lvsrv 1161
n_cst_tvsrv 1348
of_SetDisplayStyle 471
of_SetDisplayUnits
n_cst_dwsrv 720
n_cst_lvsrv 1162
n_cst_tvsrv 1349
of_SetDropDown
u_calculator 410
u_calendar 447
of_SetDropDownCalculator
u_dw 181
u_em 205
of_SetDropDownCalendar
u_dw 182
u_em 205
of_SetDropDownSearch 183
of_SetDwCache 598
of_SetDWProperty 667
of_SetEnabled
n_cst_dwsrv_printpreview 851
n_cst_dwsrv_querymode 866
of_SetError 599
of_SetExclude
n_cst_dwsrv_filter 744
n_cst_dwsrv_sort 983
n_cst_lvsrv_sort 1208
of_SetFileArchive 1044
of_SetFileAttributes
n_cst_filesrv 1045
n_cst_filesrvmac 1068
n_cst_filesrvsol2 1090
n_cst_filesrvwin16 1107
n_cst_filesrvwin32 1129
of_SetFileHidden 1046
of_SetFileName 298
of_SetFileReadonly 1046
of_SetFileSystem 1047
of_SetFillColor 472
of_SetFillStyle 472
of_SetFilter
n_cst_dwsrv_filter 744
u_dw 183
of_SetFind
u_dw 184
u_rte 299
of_SetFocusOnRequestor
u_calculator 411
u_calendar 447
of_SetFont
n_cst_dwsrv_report 909
u_progressbar 473
of_SetFontBold 475
of_SetFontCharSet 476
of_SetFontFace 476
of_SetFontFamily 477
of_SetFontItalic 478
of_SetFontPitch 478
of_SetFontSize 479
of_SetFontUnderline 479
of_SetFrame 599
of_SetGapWidth 1476
of_SetGDI 1477
of_SetGDIThreshold 1477
of_SetGDIWidth 1478
of_SetHelpFile 600
of_SetHoliday 448
of_SetHolidayBold 449
of_SetHolidayColor 449
of_SetHorizontalPointer 497
of_SetIgnoreFileExists 300
of_SetInitialValue
u_calculator 411

u_calendar 450
of_SetItem 721
of_SetItemAttributes
n_cst_lvsrv_datasource 1196
n_cst_tvsrv_levelsource 1394
of_SetKeyCols 813
of_SetKeyValue 1331
of_SetKeyValues 813
of_SetLastAccessDate
n_cst_filesrv 1047
n_cst_filesrvmac 1069
n_cst_filesrvsol2 1091
n_cst_filesrvwin32 1130
of_SetLastWriteDateTime
n_cst_filesrv 1048
n_cst_filesrvsol2 1092
n_cst_filesrvwin16 1108
n_cst_filesrvwin32 1131
of_SetLineEnding 1137
of_SetLinkage 184
of_SetLogFile
n_cst_error 1010
n_cst_sqlspy 1310
of_SetLogFileStyle 1011
of_SetLogicalUnitOfWork
u_base 385
u_lvs 245
u_tab 325
u_tvs 363
w_master 50
of_SetLogo 600
of_SetLogSeverity 1012
of_SetMarkedDay 451
of_SetMarkedDayBold 451
of_SetMarkedDayColor 452
of_SetMaster 814
of_SetMaximum 480
of_SetMem 1478
of_SetMemThreshold 1479
of_SetMemWidth 1479
of_SetMenuItems 1421
of_SetMessageText 480
of_SetMicrohelp 601
of_SetMinimum 481
of_SetMinObjectSize 497
of_SetMinSize
n_cst_dwsrv_resize 939
n_cst_resize 1284
of_SetMRU 601
of_SetMultiTable
n_ds 527
u_dw 185
of_SetName 567
of_SetNotifyConnection 1012
of_SetNotifyPreTitle 1013
of_SetNotifySeverity 1013
of_SetNotifyWho 1014
of_SetOrigSize
n_cst_dwsrv_resize 940
n_cst_resize 1285
of_SetOtherSaveObjects 815
of_SetOverlayPicture
n_cst_lvsrv_datasource 1179
n_cst_tvsrv_levelsource 1370
of_SetOverLayPictureColumn
n_cst_lvsrv_datasource 1197
n_cst_tvsrv_levelsource 1394
of_SetPageInputField 300
of_SetParent
m_dw 67
m_edit 70
m_oc 83
m_tvs 85
m_view 74
of_SetPictureColumn
n_cst_lvsrv_datasource 1198
n_cst_tvsrv_levelsource 1395
of_SetPosition 482
of_SetPosSize 1421
of_SetPredefinedSource 1015
of_SetPreference 50
of_SetPrint 363
of_SetPrintPreview
n_ds 528
u_dw 185
of_SetProperty 186
of_SetQueryCols 867
of_SetQuerymode 187
of_SetRecursive 1396
of_SetRedraw 815
of_SetRefreshRate 1480
of_SetRelativeZoom 913

of_SetReport
n_ds 528
u_dw 187
of_SetReqColumn 188
of_SetRequestor
n_cst_apppreference 622
n_cst_dropdown 674
n_cst_dwsrv 721
n_cst_dwsrv_resize 941
n_cst_luw 1151
n_cst_lvsrv 1162
n_cst_rtefind 1292
n_cst_tvsrv 1349
n_cst_winsrv 1406
u_calculator 412
u_calendar 453
of_SetResetCriteria 868
of_SetResize
u_base 385
u_dw 188
u_tab 326
w_master 51
of_SetRestoreApp 622
of_SetRestoreRow 952
of_SetRestoreUser 623
of_SetRetrieveOnDisabled 868
of_SetRmbMenu 364
of_SetRowManager 189
of_SetRowSelect 189
of_SetRuler 852
of_SetSaturdayBold 454
of_SetSaturdayColor 454
of_SetSaveSound 816
of_SetSecurity
n_cst_appmanager 602
n_cst_security 1297
of_SetSelectedPictureColumn 1397
of_SetSharedProperty 190
of_SetSheetManager 18
of_SetSort
n_cst_dwsrv_sort 984
u_dw 191
u_lvs 246
of_SetSQLSpy 668
of_SetStartPageNumber 301
of_SetStatePictureColumn
n_cst_lvsrv_datasource 1199
n_cst_tvsrv_levelsource 1398
of_SetStatusBar 18
of_SetStep 482
of_SetStyle
n_cst_dwsrv_filter 745
n_cst_dwsrv_linkage 817
n_cst_dwsrv_rowselection 970
n_cst_dwsrv_sort 985
n_cst_error 1016
of_SetSundayBold 455
of_SetSundayColor 455
of_SetSyncOnKeyChange 817
of_SetTextColor 483
of_SetTextStyleBold 301
of_SetTextStyleItalic 302
of_SetTextStyleStrikeout 302
of_SetTextStyleSubscript 303
of_SetTextStyleSuperscript 304
of_SetTextStyleUnderline 304
of_SetTimeOut 1017
of_SetTimer 1480
of_SetTimerFormat 1481
of_SetTimerInterval 1482
of_SetTimerWidth 1482
of_SetToNullIfEmpty 922
of_SetToolbarItemOrder 1422
of_SetToolbarItemSpace 1423
of_SetToolbarItemVisible 1423
of_SetToolbars 1424
of_SetToolbarTitles 1424
of_SetTrace 567
of_SetTransObject
n_cst_dwsrv_linkage 818
n_ds 529
u_dw 191
of_SetTrRegistration 602
of_SetTypeToProcess 1152
of_SetUnattended 1017
of_SetUpdateable
n_ds 529
u_base 386
u_dw 192
u_lvs 246
u_tab 326
u_tvs 364

w_master 52
of_SetUpdateBottomUp 819
of_SetUpdateObjects
u_base 386
u_tab 327
w_master 52
of_SetUpdateOnRowChange 819
of_SetUpdateRequestor
n_cst_dwsrv_linkage 820
n_cst_luw 1153
n_ds 530
u_base 387
u_dw 192
u_lvs 247, 327
u_tvs 365
w_master 53
of_SetUpdateStyle 821
of_SetUseColLinks 822
of_SetUseDisplay 986
of_SetUser
n_cst_error 1018
n_cst_winsrv_statusbar 1483
n_tr 568
of_SetUserID 603
of_SetUserIniFile
n_cst_appmanager 603
n_cst_apppreference 624
of_SetUserKey
n_cst_appmanager 604
n_cst_apppreference 625
of_SetUserThreshold 1483
of_SetUserWidth 1484
of_SetValue 413
of_SetValueOnRequestor 414
of_SetVersion 604
of_SetVerticalPointer 498
of_SetVisibleOnly
n_cst_dwsrv_filter 746
n_cst_dwsrv_sort 986
of_SetWindow 1425
of_SetXPosColumn 1199
of_SetYPosColumn 1200
of_SetZoom 852
of_ShiftBand 914
of_Sort 987
of_SortDirList 1048
of_Splash 605
of_SQLPreviewType 635
of_SQLSyntax 1310
of_StoreUpdateSettings 841
of_String 636
of_Time 639
of_ToolbarAlignment 640
of_ToolbarExists 1214
of_TriggerEvent
n_cst_dwsrv_linkage 822
n_cst_menu 1215
of_Trim 1331
of_TypeOf 1285
of_UnDelete 953
of_Undo 915
of_Unlink 823
of_UnlinkDetail 823
of_UnRegister
n_cst_dwcache 693
n_cst_dwsrv_dropdownsearch 733
n_cst_dwsrv_linkage 824
n_cst_dwsrv_multitable 842
n_cst_dwsrv_reqcolumn 923
n_cst_dwsrv_resize 941
n_cst_resize 1286
n_cst_trregistration 1339
n_cst_winsrv_statusbar 1484
u_calculator 414
u_calendar 456
of_UnRegisterPredefined 1485
of_UnRegisterReportColumn 1201
of_UnRegister 1399
of_Update
n_cst_dwsrv_linkage 824
n_cst_dwsrv_multitable 843
n_cst_luw 1153
n_cst_lvsrv_datasource 1201
n_cst_tvsrv_levelsource 1399
n_ds 530
u_base 387
u_dw 193
u_lvs 247
u_tab 328
u_tvs 365
w_master 53
of_UpdateBottomUp 825

of_UpdateBottomUpAndTopDown 826
of_UpdateChecks 54
of_UpdateCustom 827
of_UpdateObjectData 499
of_UpdateOnRowChange 828
of_UpdatePredefined 1486
of_UpdatePrep 829
  n_cst_luw 1154
  n_ds 533
  u_base 388
  u_dw 195
  u_lvs 248
  u_tab 329
  u_tvs 366
  w_master 55
of_UpdateSkipString 924
of_UpdatesPending
  n_cst_luw 1155
  n_ds 533
  u_base 389
  u_dw 196
  u_lvs 249
  u_tab 329
  u_tvs 367
  w_master 56
of_UpdateTopDown 829
of_UpdateTopDownAndBottomUp 830
of_UpdateVisuals 483
of_UsesDisplayValue 988
of_Validation
  n_cst_dwsrv_linkage 831
  n_cst_luw 1155
  n_ds 534
  u_base 390
  u_dw 197
  u_lvs 249
  u_tab 330
  u_tvs 368
  w_master 56
of_Wait 656
of_WeekNumber 658
of_WeeksAfter 659
of_WindowState 641
of_WordCap 1332
of_YearsAfter 659
OLEControl 257
OLEObject object 544
OLEStorage object 545
OLEStream object 546
online Help
  accessing 9
  of_GetHelpFile 590
  of_SetHelpFile 600
Open event 25

P

PEAT sample application 10
pfc_AcceptText
  u_base 376
  u_dw 154
  u_lvs 223
  u_tab 316
  u_tvs 336
  w_master 25
pfc_AddAll
  u_lvs 223
  u_tvs 336
pfc_AddItem
  u_lvs 224
  u_tvs 337
pfc_AddRow
  n_cst_dwsrv_rowmanager 944
  u_dw 154
pfc_Apply 60
pfc_BeginTran 25
pfc_Cancel 60
pfc_Cascade
  n_cst_winsrv_sheetmanager 1428
  w_frame 15
pfc_Clear
  u_ddlb 134
  u_ddplb 141
  u_dw 154
  u_em 200
  u_mle 252
  u_oc 258
  u_rte 282
  u_sle 307
pfc_Clicked
  n_cst_dwsrv_linkage 770

n_cst_dwsrv_rowselection 957
n_cst_dwsrv_sort 974
pfc_Close
n_cst_appmanager 581
n_cst_apppreference 610
w_master 25
pfc_ColumnClick 1204
pfc_ConnectionBegin 581
pfc_ConnectionEnd 582
pfc_ControlGotFocus
w_master 26
w_sheet 63
pfc_Copy
u_ddlb 134
u_ddplb 141
u_dw 155
u_em 200
u_mle 253
u_oc 258
u_rte 283
u_sle 308
pfc_Cut
u_ddlb 135
u_ddplb 142
u_dw 155
u_em 200
u_mle 253
u_oc 259
u_rte 283
u_sle 308
pfc_DBError 26
pfc_DDCalculator
u_dw 155
u_em 200
pfc_DDCalendar
u_dw 156
u_em 201
pfc_Debug 156
pfc_Default 61
pfc_Delete
u_lvs 224
u_tvs 338
pfc_DeleteItem
n_cst_lvsrv_datasource 1167
n_cst_tvsrv_levelsource 1354
pfc_DeleteRow
n_cst_dwsrv_linkage 770
n_cst_dwsrv_rowmanager 945
u_dw 157
pfc_Descendant
u_dw 157
w_master 26
pfc_DropDown
u_calculator 395
u_calendar 422
pfc_EditChanged 725
pfc_EditObject 259
pfc_EndLabelEdit
n_cst_lvsrv_datasource 1167
n_cst_tvsrv_levelsource 1354
pfc_EndTran 26
pfc_Exit 583
pfc_FilterDlg
n_cst_dwsrv_filter 737
u_dw 157
pfc_FindDlg
n_cst_dwsrv_find 749
n_cst_rtefind 1288
u_dw 157
u_rte 283
pfc_FindNext
n_cst_dwsrv_find 749
n_cst_rtefind 1288
pfc_FirstPage
u_dw 157
u_rte 284
pfc_Help 27
pfc_Hide event 1440
pfc_Idle 583
pfc_InsertFile 284
pfc_InsertItem
u_lvs 224
u_tvs 338
pfc_InsertObject 260
pfc_InsertPicture 284
pfc_InsertRow
n_cst_dwsrv_linkage 771
n_cst_dwsrv_rowmanager 945
u_dw 158
pfc_InvertSelection
u_lb 214

u_lvs 226
u_plb 271
pfc_ItemChanged 771
pfc_ItemFocusChanged
n_cst_dwsrv_dropdownsearch 725
n_cst_dwsrv_linkage 772
pfc_LastPage
u_dw 158
u_rte 284
pfc_Layer
n_cst_winsrv_sheetmanager 1429
w_frame 15
pfc_LButtonDown 957
pfc_LButtonUp 958
pfc_Logon 583
pfc_MessageRouter 28
pfc_MicroHelp
w_frame 15
w_master 28
w_sheet 63
pfc_MinimizeAll
n_cst_winsrv_sheetmanager 1429
w_frame 16
pfc_Move event 1440
pfc_MRUClicked 29
pfc_MRUProcess 29
pfc_MRURestore 30
pfc_MRUSave 30
pfc_New
u_lvs 226
u_tvs 339
w_master 30
pfc_NextDay 423
pfc_NextMonth 423
pfc_NextPage
u_dw 158
u_rte 285
pfc_NextWeek 423
pfc_Open
n_cst_appmanager 584
n_cst_apppreference 610
u_rte 285
w_master 31
pfc_OpenObject 260
pfc_Operators
n_cst_dwsrv_querymode 860
u_dw 158
pfc_PageSetup
n_ds 514
u_dw 159
w_master 31
pfc_PageSetupDlg
n_ds 514
u_dw 159
pfc_Paste
u_ddlb 135
u_ddplb 142
u_dw 159
u_em 201
u_mle 253
u_oc 260
u_rte 285
u_sle 308
pfc_PasteSpecial 261
pfc_Populate
u_lvs 226
u_tvs 339
pfc_PopulateDDDW 160
pfc_PostOpen
w_frame 16
w_master 32
pfc_PostUpdate
n_ds 515
u_base 376
u_dw 160
u_lvs 227
u_tab 317
u_tvs 340
w_master 32
pfc_PreAbout 585
pfc_PreClose 33
pfc_PreDebug 163
pfc_PreDelete
u_lvs 227
u_tvs 340
pfc_PreDeleteRow
n_cst_dwsrv_linkage 773
u_dw 161
pfc_PreFindDlg 161
pfc_PreInsert
u_lvs 227
u_tvs 341

pfc_PreInsertRow
n_cst_dwsrv_linkage 773
u_dw 161
pfc_PreLogonDlg 585
pfc_PreMRUSave 33
pfc_PreOpen 34
pfc_PrePageSetupDlg
n_ds 515
u_dw 162
pfc_PrePrintDlg
n_ds 515
u_dw 162
u_rte 286
pfc_PreRefreshItem
u_lvs 228
u_tvs 341
pfc_PreRefreshLevel 342
pfc_PreReplaceDlg 163
pfc_PreRestoreRow 164
pfc_PreRMBMenu 165
pfc_PreRmbMenu
u_ddlb 135
u_ddplb 142
u_em 201
u_lvs 228
u_mle 253
u_oc 261
u_rte 286
u_sle 308
u_tvs 342
pfc_PreSplash 586
pfc_PreToolbar 16
pfc_PreUpdate
u_dw 165
w_master 34
pfc_PrevDay 424
pfc_PreviousPage
u_dw 165
u_rte 286
pfc_PrevMonth 424
pfc_PrevWeek 424
pfc_Print
n_ds 516
u_dw 166
u_rte 287
w_master 35
pfc_PrintDlg
n_ds 516
u_dw 166
u_rte 287
pfc_PrintImmediate
n_ds 517
u_dw 166
u_rte 287
w_frame 36
pfc_PrintPreview
n_ds 517
u_dw 167
u_rte 288
pfc_Properties
u_lvs 229
u_tvs 343
pfc_RButtonDown
n_cst_dwsrv_linkage 773
n_cst_dwsrv_rowselection 959
pfc_RButtonUp 959
pfc_Refresh 229
pfc_RefreshItem
u_lvs 229
u_tvs 343
pfc_RefreshLevel 343
pfc_Rename
u_lvs 230
u_tvs 344
pfc_Replace
n_cst_dwsrv_find 750
n_cst_rtefind 1289
pfc_ReplaceAll
n_cst_dwsrv_find 750
n_cst_rtefind 1289
pfc_ReplaceDlg
n_cst_dwsrv_find 750
n_cst_rtefind 1290
u_dw 167
u_rte 288
pfc_ResetUpdate
n_ds 517
u_dw 167
pfc_Resize
n_cst_dwsrv_resize 928
n_cst_resize 1279
pfc_Resize event 1441

pfc_RestoreRow
n_cst_dwsrv_rowmanager 945
u_dw 167
pfc_Retrieve
n_ds 517
u_dw 168
u_lvs 230
u_tvs 344
pfc_RetrieveDDDW 168
pfc_RetrieveEnd 774
pfc_RetrieveStart 775
pfc_RowChanged
n_cst_dwsrv_linkage 775
u_dw 168
pfc_RowFocusChanged
n_cst_dwsrv_linkage 775
n_cst_dwsrv_rowselection 960
pfc_RowFocusChanging 776
pfc_RowValidation 169
pfc_Ruler
n_ds 518
u_dw 169
u_rte 288
pfc_Save
u_rte 288
w_master 36
pfc_SaveAs
u_rte 289
w_master 37
pfc_SaveObjects 37
pfc_SearchCompare 345
pfc_SelectAll
u_ddlb 135
u_ddplb 143
u_dw 169
u_em 202
u_lb 214
u_lvs 230
u_mle 254
u_plb 272
u_rte 289
u_sle 309
pfc_SetItemAttributes
u_lvs 231
u_tvs 345
pfc_Show event 1441
pfc_Sort 1205
pfc_SortDlg
n_cst_dwsrv_sort 975
u_dw 170
pfc_StatusBarClicked event 1442
pfc_StatusBarDoubleClick event 1442
pfc_StatusBarRButtonUp event 1443
pfc_SystemError 587
pfc_TileHorizontal
n_cst_winsrv_sheetmanager 1429
w_frame 17
pfc_TileVertical
n_cst_winsrv_sheetmanager 1429
w_frame 17
pfc_Toolbars 17
pfc_Undo
n_cst_lvsrv_datasource 1168
n_cst_tvsrv_levelsource 1355
u_dw 170
u_em 202
u_lvs 231
u_mle 254
u_rte 289
u_sle 309
u_tvs 346
pfc_UndoArrange
n_cst_winsrv_sheetmanager 1429
w_frame 17
pfc_UndoDelete
n_cst_lvsrv_datasource 1168
n_cst_tvsrv_levelsource 1356
pfc_UndoEdit
n_cst_lvsrv_datasource 1169
n_cst_tvsrv_levelsource 1356
pfc_UndoInsert
n_cst_lvsrv_datasource 1170
n_cst_tvsrv_levelsource 1357
pfc_Update
n_ds 518
u_base 377
u_dw 170
u_lvs 231
u_tab 317
u_tvs 346
w_master 38
pfc_UpdateLinks 261

pfc_UpdateObjects 38
pfc_UpdatePrep
  n_ds 518
  u_base 377
  u_dw 171
  u_lvs 232
  u_tab 318
  u_tvs 347
  w_master 39
pfc_UpdatesPending
  n_ds 519
  u_base 378
  u_dw 171
  u_lvs 232
  u_tab 318
  u_tvs 347
  w_master 39
pfc_UpdatesPendingRef 40
pfc_Validation
  n_ds 519
  u_base 378
  u_dw 171
  u_lvs 233
  u_tab 319
  u_tvs 348
  w_master 41
pfc_Values
  n_cst_dwsrv_querymode 860
  u_dw 171
pfc_Zoom
  n_ds 519
  u_dw 171
PFCMacPlatformSrv 1239
Picture 265
PictureButton 268
PictureListBox 271
Pipeline object 547
platform service
  f_SetPlatform 89
  n_cst_platform 1230
  n_cst_platformaix 1246
  n_cst_platformhpux 1246
  n_cst_platformmac 1239
  n_cst_platformsol2 1246
  n_cst_platformwin16 1256
  n_cst_platformwin32 1266
popup menu
  m_dw 66
  m_edit 69
  m_lvs 73
  m_oc 82
  m_tvs 84
Powersoft Online Books 9
preference service 1408
print preview service 844
PrintHeader event 289
property service 854

Q

query mode service 858

R

RadioButton 276
RButtonDown event
  u_dw 172
  u_progressbar 460
RButtonUp event
  u_ddlb 136
  u_ddplb 143
  u_dw 172
  u_em 202
  u_lvs 233
  u_mle 254
  u_oc 262
  u_rte 290
  u_sle 309
  u_tvs 348
registry
  of_GetAppKey 589
  of_IsRegistryAvailable 594
  of_SetAppKey 596, 621
  of_SetUserKey 604, 625
report service
  n_cst_dssrv_report 678
  n_cst_dwsrv_report 870
required column 917
Resize event
  u_base 379

u_dw 172
u_tab 319
w_frame 17
w_master 41
resize service
DataWindow 925
n_cst_resize 1276
RetrieveEnd event 172
RetrieveStart event
n_ds 519
u_dw 173
RichTextEdit 279
RightClicked event
u_lvs 233
u_tvs 348
Rollback 565
row manager service 943
row selection services 954
RowFocusChanged event 173

S

s_pagesetupattrib 121
s_paperattrib 122
s_printdlgattrib 123
s_svalue 124
sample application 10
security service
n_cst_security 1294
of_SetAppPreference 597
of_SetMRU 601
of_SetSecurity 602
selection service 1298
Service object 549
sheet manager service 1427
Show event 18
SingleLineEdit 306
Sort event 233
sort service 972
splash screen
of_Splash 605
pfc_PreSplash 586
splitbar service 485
SQL parsing service 1303
SQL Spy
main discussion 1306
of_SetSQLSpy 668
SQLPreview event
n_ds 520
u_dw 173
StaticText 312
status bar service 1435
string parsing service 1312

T

Tab control 315
TabPage 501
token 1315
Transaction object 553
transaction registration service
main discussion 1334
of_SetTrRegistration 602
Transport object 569
TreeView 331
TreeView base service 1341
TreeView level source service 1351
TreeView print service 1401

U

u_base 374
u_calculator 391
u_calendar 416
u_cb 126
u_cbx 129
u_ddlb 132
u_ddplb 139
u_dw
m_dw 66
main discussion 146
u_em 198
u_gb 206
u_gr 207
u_hsb 210
u_lb 213
u_lvs
m_lvs 73
main discussion 218

u_mle 251
u_oc
  m_oc 82
  main discussion 257
u_p 265
u_pb 268
u_plb 271
u_progressbar 457
u_rb 276
u_rte
  main discussion 279
  n_cst_rtefind 1287
u_sle 306
u_st 312
u_st_splitbar 485
u_tab
  main discussion 315
  n_cst_resize 1276
u_tabpg 501
u_tvs
  m_tvs 84
  main discussion 331
u_vsb 369
user INI file 603, 624

V

VerticalScrollBar 369

W

w_child 12
w_debuglog 666
w_frame 13
w_main 20
w_master
  main discussion 21
  n_cst_resize 1276
w_popup 58
w_response 59
w_sheet 62
w_sqlspy 1308
window services
  n_cst_winsrv 1405
  n_cst_winsrv_preference 1408
  n_cst_winsrv_sheetmanager 1427
  n_cst_winsrv_statusbar 1435

Z

zoom
  of_GetZoom 848
  of_SetRelativeZoom 913
  pfc_Zoom (n_ds) 519
  pfc_Zoom (u_dw) 171