



PowerBuilder Foundation Class Library Object Reference Volume 2

DOCUMENT ID: 37786-01-0900-01

LAST REVISED: March 2003

Copyright © 1989-2003 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. 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.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-FORMS, APT-Translator, APT-Library, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers' Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, MainframeConnect, Maintenance Express, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, PB-Gen, PC APT Execute, PC Net Library, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, Report Workbench, Report-Execute, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Resource Manager, RW-DisplayLib, S-Designer, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TradeForce, Transact-SQL, Translation Toolkit, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, Viewer, Visual Components, VisualSpeller, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 11/02

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

VOLUME 2

CHAPTER 9	Custom Class User Objects	551
	n_base	552
	n_cst_appmanager	554
	n_cst_apppreference	584
	n_cst_color	602
	n_cst_conversion	604
	n_cst_datetime	616
	n_cst_debug	634
	n_cst_dropdown	641
	n_cst_dssrv	647
	n_cst_dssrv_multitable	648
	n_cst_dssrv_printpreview	649
	n_cst_dssrv_report	650
	n_cst_dwcache	651
	n_cst_dwsrv	665
	n_cst_dwsrv_dropdownsearch	689
	n_cst_dwsrv_filter	700
	n_cst_dwsrv_find	711
	n_cst_dwsrv_linkage	727
	n_cst_dwsrv_multitable	790
	n_cst_dwsrv_printpreview	802
	n_cst_dwsrv_property	811
	n_cst_dwsrv_querymode	815
	n_cst_dwsrv_report	826
	n_cst_dwsrv_reqcolumn	869
	n_cst_dwsrv_resize	877
	n_cst_dwsrv_rowmanager	894
	n_cst_dwsrv_rowselection	904
	n_cst_dwsrv_sort	920
	n_cst_error	936
	n_cst_filesrv	964
	n_cst_filesvrwin32	990

n_cst_infile	1012
n_cst_linkedListbase	1017
n_cst_linkedListnode	1025
n_cst_linkedListnodecompare	1026
n_cst_list	1029
n_cst_luw	1038
n_cst_lvsvr	1054
n_cst_lvsvr_datasource	1060
n_cst_lvsvr_sort	1102
n_cst_menu.....	1108
n_cst_metaClass	1114
n_cst_mru	1121
n_cst_nodebase	1143
n_cst_nodeComparebase	1149
n_cst_numerical	1152
n_cst_platform.....	1158
n_cst_platformWin32	1165
n_cst_queue.....	1174
n_cst_resize	1177
n_cst_rteFind.....	1187
n_cst_security	1193
n_cst_selection	1197
n_cst_sql	1202
n_cst_sqLSPY	1205
n_cst_stack	1211
n_cst_string.....	1214
n_cst_tmGmMultiple	1234
n_cst_tmGsSingle	1244
n_cst_tree	1253
n_cst_treenode	1266
n_cst_treenodeCompare	1270
n_cst_trRegistration.....	1272
n_cst_tvSrv	1281
n_cst_tvSrv_LevelSource	1290
n_cst_tvSrv_Print	1347
n_cst_winSrv	1351
n_cst_winSrv_Preference	1354
n_cst_winSrv_SheetManager	1372
n_cst_winSrv_StatusBar.....	1379
Index	1429

Custom Class User Objects

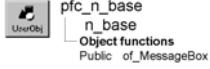
About this chapter

This chapter describes the custom class user objects in PFC.

Contents

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

n_base

Description	Ancestor object for all PFC custom class user objects.
Ancestry	 <pre>graph TD; pfc_n_base[pfc_n_base] --> n_base[n_base]; n_base --> UserObj[UserObj]</pre>
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	<code>instancename.of_MessageBox (id, title, message, icon, button, default)</code>
Argument	Description
<i>instancename</i>	Instance name of n_base
<i>id</i>	String specifying an ID for the message
<i>title</i>	String specifying the title of the MessageBox
<i>message</i>	String specifying the message text
<i>icon</i>	Icon enumerated data type indicating the icon you want to display on the left side of the message box. Values are: <ul style="list-style-type: none">• Information!• StopSign!• Exclamation!• Question!• None!
<i>button</i>	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: <ul style="list-style-type: none">• OK!• OKCancel!• YesNo!• YesNoCancel!• RetryCancel!• AbortRetryIgnore!
<i>default</i>	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: <ol style="list-style-type: none">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).<pre>n_cst_appmanager gnv_app</pre>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:<pre>gnv_app = CREATE n_cst_appmanager gnv_app.Event pfc_Open(commandline)</pre>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_SetSecurity5 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
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_apppref	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 image 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 pre-coded 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
<code>is_appinifile</code>	<code>of_SetAppIniFile</code>
<code>is_appkey</code>	<code>of_SetAppKey</code>
<code>is_appinifile</code>	<code>of_SetAppIniFile</code>
<code>is_helpfile</code>	<code>of_SetHelpFile</code>
<code>is_userinifile</code>	<code>of_SetUserIniFile</code>
<code>is_userkey</code>	<code>of_SetUserKey</code>
<code>is_version</code>	<code>of_SetVersion</code>

Examples This example initializes selected instance variables:

```
of_SetAppIniFile("c:\eis\veisapp.ini")
of_SetUserIniFile("c:\eis\veisuser.ini")
of_SetHelpFile("c:\eis\veis.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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>userid</i>	String specifying the user ID. From within the pfc_ConnectionBegin event, access this value through the <i>as_userid</i> argument
<i>password</i>	String specifying the password. From within the pfc_ConnectionBegin event, access this value through the <i>as_password</i> argument

	Argument	Description
	<i>connectstring</i>	String containing the connectstring. From within the pfc_ConnectionBegin event, access this value through the <i>as_connectstring</i> argument
Return value		ConnectPrivilege. Returns one of the following: ConnectPrivilege! ConnectWithAdminPrivilege! NoConnectPrivilege!
Usage		To use this event: <ol style="list-style-type: none"> 1 Call pfc_ConnectionBegin from the application object's ConnectionBegin event, passing the <i>userid</i>, <i>password</i>, and <i>connectstring</i> 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. <p>For more information on distributed computing, see <i>Application Techniques</i>.</p>
Examples		This example returns the client's connection privilege:
		<pre>CHOOSE CASE as_userid CASE "adm" Return ConnectWithAdminPrivilege! CASE "ctl", "mmw", "jdb" Return ConnectPrivilege! CASE ELSE Return NoConnectPrivilege! END CHOOSE</pre>

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: <ol style="list-style-type: none"> 1 Call pfc_ConnectionEnd from the application object's ConnectionEnd event. 2 Add logic to pfc_ConnectionEnd.

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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>userid</i>	String specifying the user ID. From within the pfc_Logon event, access this value through the <i>as_userid</i> argument
<i>password</i>	String specifying the password. From within the pfc_Logon event, access this value through <i>as_password</i> 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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>commandline</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>attributes</i>	N_cst_aboutattrib variable into which the event places application, logo, version, and copyright information. This argument is accessed through the <i>anv_aboutattrib</i> 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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>attributes</i>	N_cst_logonattrib variable into which the event places user ID, application, and logo information. This argument is accessed through the <i>anv_logonattrib</i> 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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>attributes</i>	N_cst_splashattrib variable into which the event places application, logo, version, and copyright information. This argument is accessed through the <i>anv_splashattrib</i> 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 pre-coded 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	<i>instancename.of_About ()</i>				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 2px;">Argument</th><th style="text-align: center; padding: 2px;">Description</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"><i>instancename</i></td><td style="padding: 2px;">Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
Argument	Description				
<i>instancename</i>	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_AppIniFile

Description	Retrieves the name of the application INI file as established by the of_SetAppIniFile function.				
Access	Public				
Syntax	<i>instancename.of_AppIniFile ()</i>				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 2px;">Argument</th><th style="text-align: center; padding: 2px;">Description</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;"><i>instancename</i></td><td style="padding: 2px;">Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
Argument	Description				
<i>instancename</i>	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_AppIniFile function to retrieve database connection information:				

```
SQLCA.dbms=ProfileString &
(gnv_app.of_AppIniFile(), &
"Database", "DBMS", " ")
SQLCA.database=ProfileString &
(gnv_app.of_AppIniFile(), &
"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_AppKey

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

Access Public

Syntax *instancename.of_AppKey ()*

Argument	Description
<i>instancename</i>	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_AppKey function:

```
String      ls_appkey

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

of_Copyright

Description Retrieves the copyright statement as established by the of_SetCopyright function.

Access Public

Syntax *instancename.of_Copyright ()*

Argument	Description
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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	<i>instancename.of_GetUserID()</i>	
Argument	Description	
<i>instancename</i>	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:	
	<pre>st_version.text = gnv_app.of_GetVersion() st_user.text = gnv_app.of_GetUserID()</pre>	

of.GetUserIniFile

Description	Retrieves the name of the user INI file as established by the of_SetUserIniFile function.	
Access	Public	
Syntax	<i>instancename.of.GetUserIniFile()</i>	
Argument	Description	
<i>instancename</i>	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:	
	<pre>this.inv_preference.of_Save & (gnv_app.of.GetUserIniFile(), "Position")</pre>	

of.GetUserKey

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

Syntax	<i>instancename.of_GetUserKey()</i>				
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
Argument	Description				
<i>instancename</i>	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	<i>instancename.of_GetVersion()</i>				
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
Argument	Description				
<i>instancename</i>	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	<i>instancename.of_IsRegistryAvailable()</i>				
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
Argument	Description				
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)				
Return value	Boolean. Returns TRUE if the workstation is running Windows 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.				
<hr/>					
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.				
<hr/>					
Examples	This example calls the of_IsRegistryAvailable function:				
<pre> 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) </pre>					
of_LogonDig					
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	<i>instancename</i> .of_LogonDig ()				
<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>instancename</i></td> <td>Global variable of type n_cst_appmanager (must be gnv_app)</td> </tr> </tbody> </table>		Argument	Description	<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
Argument	Description				
<i>instancename</i>	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
<i>instancename</i>	Global variable of type <i>n_cst_appmanager</i> (must be <i>gnv_app</i>)
<i>inifilename</i>	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\veisapp.ini")
```

of_SetAppKey**Description**

Specifies the registry key for application information.

Access

Public

Syntax

instancename.of_SetAppKey (registrykey)

Argument	Description
<i>instancename</i>	Global variable of type <i>n_cst_appmanager</i> (must be <i>gnv_app</i>)
<i>registrykey</i>	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 <code>is_appkey</code> instance variable.
Examples	This example calls the <code>of_SetAppKey</code> 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 <code>n_cst_apppreference</code> user object. You reference this instance of <code>n_cst_apppreference</code> through the <code>inv_apppref</code> instance variable.
Access	Public
Syntax	<code>instancename.of_SetAppPreference (boolean)</code>

Argument	Description
<code>instancename</code>	Global variable of type <code>n_cst_appmanager</code> (must be <code>gnv_app</code>)
<code>boolean</code>	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 <code>w_about</code> dialog box.
Access	Public
Syntax	<code>instancename.of_SetCopyright (copyrighttext)</code>

Argument	Description
<code>instancename</code>	Global variable of type <code>n_cst_appmanager</code> (must be <code>gnv_app</code>)
<code>copyrighttext</code>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>boolean</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>boolean</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app).
<i>boolean</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>framewindow</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>filename</i>	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\veisapp.hlp" )
```

of_SetLogo

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

Access Public

Syntax *instancename.of_SetLogo (logoname)*

Argument	Description
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>logoname</i>	String specifying the name of the image 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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>boolean</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>boolean</i>	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	<i>instancename.of_SetSecurity (boolean)</i>
Argument	Description
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>boolean</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>boolean</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>userid</i>	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 |
|---------------------|---|
| <i>instancename</i> | Global variable of type n_cst_appmanager (must be gnv_app) |
| <i>inifilename</i> | 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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>registrykey</i>	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 ls_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	<i>instancename.of_SetVersion (version)</i>

Argument	Description
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>version</i>	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
<i>instancename</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>secondsvisible</i>	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:

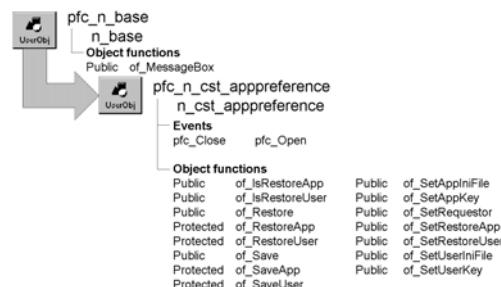
- User key
- MicroHelp
- Help file
- Version
- Logo bitmap
- Copyright notice
- DDETimeOut property
- DisplayName property
- DWMMessageTitle property
- MicrohelpDefault property
- RightToLeft property
- ToolbarFrameTitle property
- ToolbarPopMenuItemText property
- ToolbarSheetTitle property
- ToolbarUserControl 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 platforms only) or an INI file (available on all platforms).

Ancestry



Library	PFCAPSrv.PBL PFEAPSRV.PBL
Object relationships	n_cst_appattrib n_cst_conversion n_cst_infile n_cst_string
Usage	<p>Use this object to save and restore application information.</p> <p>To use n_cst_apppreference:</p> <ol style="list-style-type: none"> Enabled n_cst_apppreference by calling the n_cst_appmanager of_SetAppPreference function (this example is from an n_cst_appmanager Constructor event): <pre> this.of_SetAppPreference(TRUE)</pre> <ol style="list-style-type: none"> 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: <pre>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</pre> <ol style="list-style-type: none"> Specify the types of information to save by calling the of_SetRestoreApp and of_SetRestoreUser functions: <pre> this.inv_apppref.of_SetRestoreApp(TRUE) this.inv_apppref.of_SetRestoreUser(TRUE)</pre>
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 *applicationinstancename.EVENTpfc_Close ()*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	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.EVENTpfc_Open (cmdline)*

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

applicationinstancename.of_Restore (processapp, registrykey)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>processapp</i>	Boolean indicating whether to restore application settings (TRUE) or user settings (FALSE)
<i>registrykey</i>	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 Windows 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 *applicationinstancename.of_Restore (processapp, inifile, inisection)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>processapp</i>	Boolean indicating whether to restore application settings (TRUE) or user settings (FALSE)
<i>inifile</i>	String specifying the INI file from which to restore settings
<i>inisection</i>	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

applicationinstancename.of_RestoreApp (useregistry, keyorinifile, section)

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

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

Description

Save settings in the registry

Saves application and user settings in the registry.

Access

Public

Syntax

application.instancename.of_Save (processapp, registrykey)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>processapp</i>	Boolean indicating whether to save application settings (TRUE) or user settings (FALSE)
<i>registrykey</i>	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 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

Description

Access

Syntax

Save settings in an INI file

Saves application and user settings in an INI file.

Public

application.instancename.of_Save (processapp, infile, section)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>processapp</i>	Boolean indicating whether to save application settings (TRUE) or user settings (FALSE)
<i>infile</i>	String specifying the INI file in which to save settings
<i>section</i>	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_appnifile <> "") THEN
        li_rc = of_Save(TRUE, is_appnifile, &
                        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 *applicationinstancename.of_SaveApp (useregistry, keyorinifile, section)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is apppref)
<i>useregistry</i>	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
<i>keyorinifile</i>	String specifying the registry key or INI file in which to save settings
<i>section</i>	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 *applicationinstancename.of_SaveUser (useregistry, keyorinifile, section)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>useregistry</i>	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
<i>keyorinifile</i>	String specifying the registry key or INI file in which to save settings
<i>section</i>	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.
Access	Public
Syntax	<i>applicationinstancename.of_SetAppIniFile (inifilename)</i>

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_appref)
<i>inifilename</i>	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 platforms.

Access

Public

Syntax

applicationinstancename.of_SetAppKey (registrykey)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>registrykey</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>requestor</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>boolean</i>	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 *applicationinstancename.of_SetRestoreUser (boolean)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>boolean</i>	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.

Access

Public

Syntax

applicationinstancename.of_SetUserIniFile (inifilename)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>inifilename</i>	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 platforms.
Access	Public
Syntax	<i>applicationinstancename.of_SetUserKey (registrykey)</i>

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default for this is inv_apppref)
<i>registrykey</i>	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
    ...

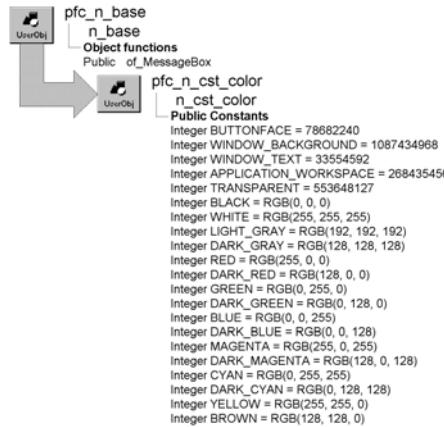
```

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:

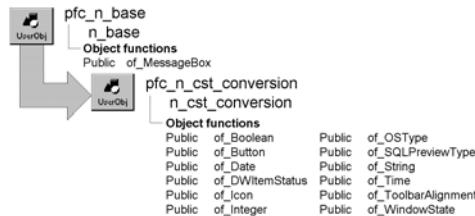
Instance variable	Description	Data type	Access	Usage
APPLICATION_WORKSPACE	Constant set to 268435456	Integer	Public	Specify colors
BLACK	Constant set to RGB(0, 0, 0)	Integer	Public	Specify colors
BLUE	Constant set to RGB(0, 0, 255)	Integer	Public	Specify colors
BROWN	Constant set to RGB(128, 128, 0)	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
<i>instancename</i>	Instance name of n_cst_conversion
<i>stringorint</i>	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
<i>instancename</i>	Instance name of n_cst_conversion
<i>string</i>	String or integer to be converted to a button. <i>String</i> must be one of the following: OK OKCancel YesNo YesNoCancel RetryCancel AbortRetryIgnore
<i>button</i>	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
<i>instancename</i>	Instance name of n_cst_conversion

	Argument	Description
	<i>string</i>	String in date or datetime format to be converted to a date data type
Return value		Date. Returns a date if the function succeeds, 1900-01-01 if <i>string</i> is an invalid date or datetime, and NULL if <i>string</i> 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 <code>of_Date</code> function:
		<pre> String ls_datetimedata = "07/28/71 12:35:14" Date ld_date Time ltm_time n_cst_conversion lnv_conversion ld_date = lnv_conversion.of_Date(ls_datetimedata) ltm_time = lnv_conversion.of_Time(ls_datetimedata) ... </pre>

of_DWItemStatus

Description Converts a string value to a dwItemStatus enumerated data type.

Access Public

Syntax `instancename.of_DWItemStatus (originalvalue, dwitemstatus)`

	Argument	Description
	<i>instancename</i>	Instance name of n_cst_conversion
	<i>originalvalue</i>	String to be converted to the dwItemStatus enumerated data type. Valid values are: DataModified New NewModified NotModified
	<i>dwitemstatus</i>	DWItemStatus enumerated data type to contain the converted string (passed by reference)
Return value		Integer. Returns 1 if the function succeeds, -1 if <i>originalvalue</i> is not a valid value, and NULL if <i>originalvalue</i> 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
<i>instancename</i>	Instance name of n_cst_conversion
<i>string</i>	String or integer to be converted to an icon. <i>String</i> must be one of the following: None Question Information StopSign Exclamation
<i>icon</i>	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[11_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
<i>instancename</i>	Instance name of n_cst_conversion
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_conversion
<i>originalvalue</i>	String to be converted to the OSType enumerated data type. Valid values are: AIX HPUX Macintosh (obsolete) OSF1 (obsolete) SOL2 Windows WindowsNT

	Argument	Description
	<i>ostype</i>	OSTypes variable into which the function returns the value (passed by reference)
Return value		Integer. Returns 1 if the function succeeds, -1 if <i>originalvalue</i> is not a valid value, and NULL if <i>originalvalue</i> 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
	<i>instancename</i>	Instance name of <i>n_cst_conversion</i>
	<i>originalvalue</i>	String to be converted to the SQLPreviewType enumerated data type. Valid values are: Insert Update Delete Retrieve
	<i>sqlpreview</i>	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

Description

Access

Syntax

To convert a boolean to a string

Creates a string from a boolean.

Public

instancename.of_String (boolean {, returntype })

Argument	Description
<i>instancename</i>	Instance name of n_cst_conversion
<i>boolean</i>	Boolean value to be converted to a string
<i>returntype</i> (optional)	<p>String specifying how the function formats the return value. Valid values are:</p> <ul style="list-style-type: none"> • 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
<i>instancename</i>	Instance name of <i>n_cst_conversion</i>
<i>input</i>	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 (obsolete)
OSF1 (obsolete)
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
<i>instancename</i>	Instance name of n_cst_conversion
<i>string</i>	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_datetimedata = "07/28/71 12:35:14"
Date       ld_date
Time       ltm_time
n_cst_conversion   lnv_conversion

ld_date = lnv_conversion.of_Date(ls_datetimedata)
ltm_time = lnv_conversion.of_Time(ls_datetimedata)
...

```

of_ToolbarAlignment

Description Converts a string value to the ToolbarAlignment enumerated data type.

Access Public

Syntax *instancename.of_ToolbarAlignment (originalvalue, toolbaralign)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_conversion
<i>originalvalue</i>	String to be converted to the SQLPreviewType enumerated data type. Valid values are: Bottom Floating Left Right Top
<i>toolbaralign</i>	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
<i>instancename</i>	Instance name of n_cst_conversion
<i>string</i>	String or integer to be converted to a windowstate. <i>String</i> must be one of the following: Maximized Minimized Normal
<i>windowstate</i>	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)

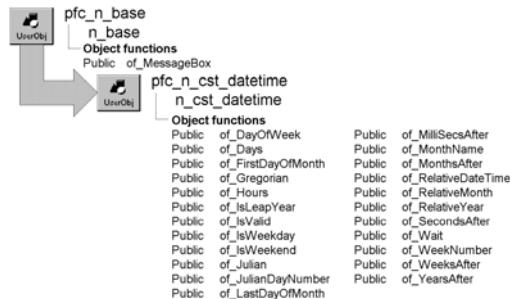
```

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
<i>instancename</i>	Instance name of n_cst_datetime
<i>sourcedate</i>	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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>seconds</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>sourcedate</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>juliandate</i>	Long containing the Julian date to be converted to Gregorian (the basis for <i>juliandate</i> is January 1, 0000)

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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>seconds</i>	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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>sourcedate</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>source</i>	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 if *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(sle_date.Text)
IF NOT lb_valid THEN
    MessageBox("Date/Time", &
               sle_date.text + " is not a valid date")
END IF
```

Date function always returns valid date

If you cast the user-entered string to Date format (`Date (sle_date.Text)`), the of_IsValid function will always return true. This is because Date will return 1900-01-01 for any invalid date.

of_IsWeekDay

Description Determines if a passed date is a weekday.

Access Public

Syntax *instancename.of_IsWeekDay (source)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_datetime
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>gregoriandate</i>	Date containing the Gregorian date to be converted to Julian (the basis for <i>gregoriandate</i> is January 1, 0000)

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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>source</i>	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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>sourcedate</i>	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 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
<i>instancename</i>	Instance name of n_cst_datetime
<i>starttime</i>	Time variable containing the starting time
<i>endtime</i>	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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>numberordate</i>	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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>startdate</i>	Date variable containing the start date
<i>enddate</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>start</i>	Datetime variable that the function uses as the starting point
<i>offset</i>	Long representing the number of seconds between <i>start</i> 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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>start</i>	Date variable that the function uses as the starting point
<i>offset</i>	Long representing the number of months between <i>start</i> 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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>start</i>	Date variable that the function uses as the starting point
<i>offset</i>	Long representing the number of years between <i>start</i> 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
<code>instancename</code>	Instance name of <code>n_cst_datetime</code>
<code>starttime</code>	Datetime variable containing the starting time
<code>endtime</code>	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

Description

Wait until a specified date/time

Stops processing until a specified date/time is reached.

Access

Public

Syntax

instancename.of_Wait (waituntil)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>waituntil</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>waitseconds</i>	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 whole weeks between the first day of the year and the specified date.

Access Public

Syntax *instancename.of_WeekNumber (date)*

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

Return value Long. Returns the number of whole weeks between the first day of the year and the specified 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
<i>instancename</i>	Instance name of <i>n_cst_datetime</i>
<i>startdate</i>	Date variable containing the start date
<i>enddate</i>	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
<i>instancename</i>	Instance name of n_cst_datetime
<i>startdate</i>	Date variable containing the start date
<i>enddate</i>	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

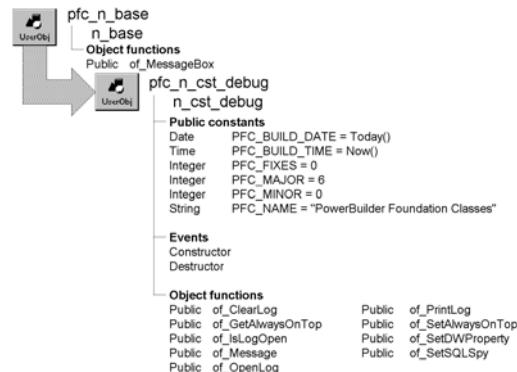
Description

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 |
|---------------------|---|
| <i>instancename</i> | 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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
<i>message</i>	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
<i>instancename</i>	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
<i>boolean</i>	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	<code>instancename.of_PrintLog()</code>				
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><code>instancename</code></td><td>Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)</td></tr> </tbody> </table>	Argument	Description	<code>instancename</code>	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
Argument	Description				
<code>instancename</code>	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
<code>instancename</code>	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
<code>boolean</code>	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
<i>instancename</i>	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
<i>boolean</i>	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)*

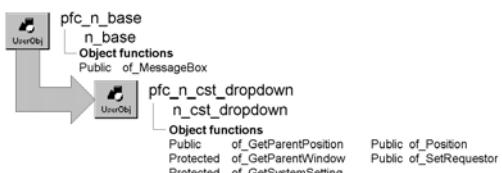
Argument	Description
<i>instancename</i>	Instance name of n_cst_debug (the n_cst_appmanager default for this is inv_debug)
<i>boolean</i>	Boolean indicating whether to start (TRUE) or stop (FALSE) the SQL Spy service

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	 <pre> graph TD pfc_n_base[pfc_n_base] --> n_base[n_base] n_base --> pfc_n_cst_dropdown[pfc_n_cst_dropdown] </pre> <p>pfc_n_base n_base Object functions Public of_MessageBox</p> <p>pfc_n_cst_dropdown n_cst_dropdown Object functions Public of_GetParentPosition Protected of_GetParentWindow Protected of_SetRequestor Public of_Position Public of_SetRequestor Protected of_GetSystemSetting</p>
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
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

Instance variable	Description	Data type	Access	Usage
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
<i>instancename</i>	Instance name of <i>n_cst_dropdown</i>
<i>control</i>	DragObject whose parent position is returned
<i>x</i>	Integer into which the function places the parent's x coordinate (passed by reference)
<i>y</i>	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_parentry)
...

```

of_GetParentWindow**Description**

Retrieves the window containing the current object.

Access

Protected

Syntax

instancename.of_GetParentWindow (control, window)

Argument	Description
<i>instancename</i>	Instance name of n_cst_dropdown
<i>control</i>	DragObject whose window is returned
<i>window</i>	Window variable into which the function places a reference to the window containing <i>control</i> . 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_parentry)
...

```

of_GetSystemSetting**Description**

Retrieves the platform-specific setting.

Access

Protected

Syntax	<i>instancename.of_GetSystemSetting (settingtype)</i>						
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of n_cst_dropdown</td></tr><tr><td><i>settingtype</i></td><td>Integer or n_cst_dropdown constant specifying the setting to be returned:<ul style="list-style-type: none">• 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</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_dropdown	<i>settingtype</i>	Integer or n_cst_dropdown constant specifying the setting to be returned: <ul style="list-style-type: none">• 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
Argument	Description						
<i>instancename</i>	Instance name of n_cst_dropdown						
<i>settingtype</i>	Integer or n_cst_dropdown constant specifying the setting to be returned: <ul style="list-style-type: none">• 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
<i>instancename</i>	Instance name of n_cst_dropdown
<i>control</i>	DragObject containing the control used to calculate the new position
<i>makevisible</i>	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
<i>instancename</i>	Instance name of n_cst_dropdown
<i>requestor</i>	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 complete information on the instance variables, events, and functions in `n_cst_dssrv`, see [n_cst_dwsrv](#) on page 665.

Descendants `n_cst_dssrv_multitable`
`n_cst_dssrv_printpreview`
`n_cst_dssrv_report`

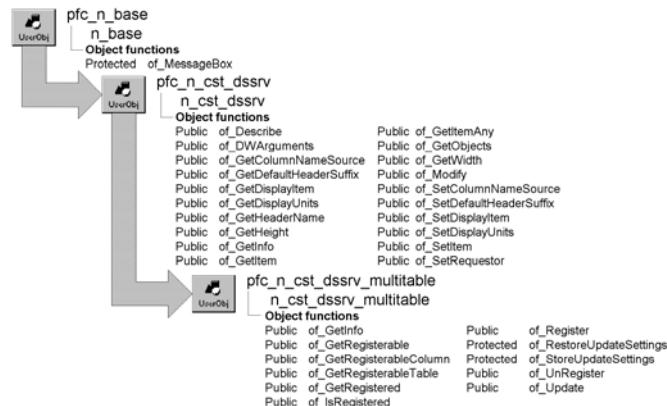
See also `n_cst_dwsrv`
`n_ds`

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 complete information on the instance variables, events, and functions in **n_cst_dssrv_multitable**, see **n_cst_dwsrv_multitable** on page 790.

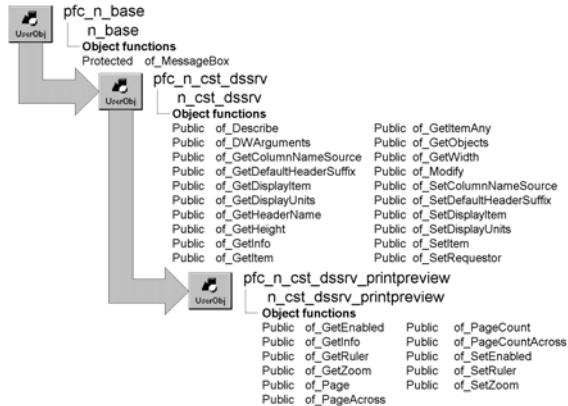
See also

[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.

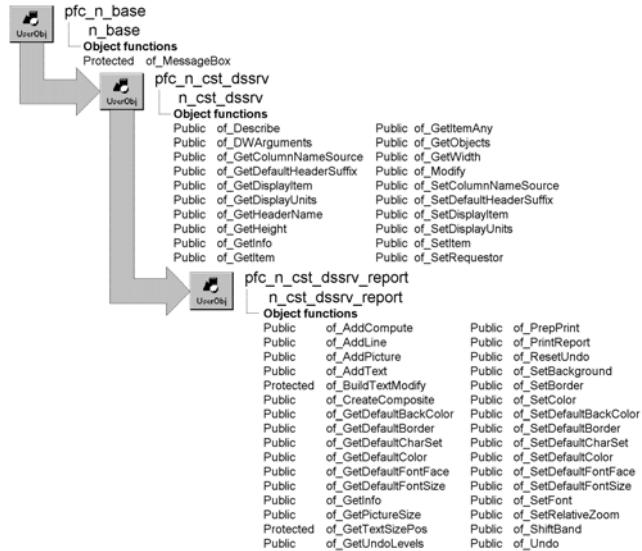
For complete information on the instance variables, events, and functions in `n_cst_dssrv_printpreview`, see `n_cst_dwsrv_printpreview` on page 802.

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 complete information on the instance variables, events, and functions in n_cst_dssrv_report, see n_cst_dwsrv_report on page 826.

See also

n_cst_dssrv

n_cst_dssrv_multitable

n_cst_dssrv_printpreview

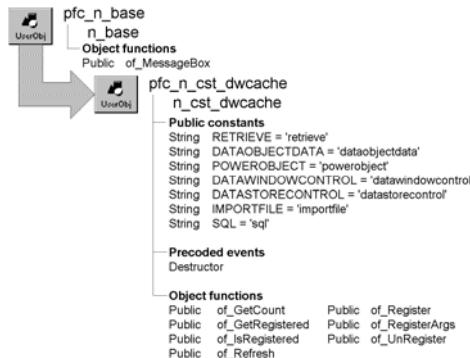
n_cst_dwsrv_report

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.

PFC 7 custom class extension

To further the extensibility of PFC, a new n_cst_dwcacheattrib custom class user object replaces the os_cachelist object structure that is now obsolete.

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
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
inv_cachelist []	Custom class user object containing cached data	n_cst_dwcache attrib	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

Instance variable	Description	Data type	Access	Usage
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 | <code>applicationinstancename.of_GetCount()</code> |

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	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

applicationinstancename.of_GetRegistered (id, cache)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	String specifying the cached DataWindow objects ID, as specified in the of_Register function
<i>cache</i>	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

`applicationinstancename.of_GetRegistered (idlist)`

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

`applicationinstancename.of_GetRegistered (dslist)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>dslist</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i> (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

applicationinstancename.of_Register (id, dwobject {, trans {, arguments {, initialload } } })

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	String identifying the cached DataWindow object
<i>dwobject</i>	String specifying the DataWindow object to be cached
<i>trans</i> (optional)	N_tr transaction object used to retrieve rows for <i>dwobject</i> If you do not specify this argument, <i>dwobject</i> must have been saved with the data
<i>arguments</i> (optional)	Twenty-element array of the Any data type specifying arguments for <i>dwobject</i>
<i>initialload</i> (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

Description	To cache data retrieved via a DataWindow object Retrieves rows from the database (using the specified DataWindow object) and caches them.														
Access	Public														
Syntax	<code>application.instancename.of_Register (id, dwobject {, trans {, initialload } })</code>														
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>application</i></td><td>Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)</td></tr> <tr> <td><i>id</i></td><td>String identifying the cached DataWindow object</td></tr> <tr> <td><i>dwobject</i></td><td>String specifying the DataWindow object to be cached</td></tr> <tr> <td><i>trans</i> (optional)</td><td>N_tr transaction object used to retrieve rows for <i>dwobject</i> If you do not specify this argument, <i>dwobject</i> must have been saved with the data</td></tr> <tr> <td><i>initialload</i> (optional)</td><td>Boolean specifying whether to load the data immediately or wait until the first of_Refresh request</td></tr> </tbody> </table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)	<i>id</i>	String identifying the cached DataWindow object	<i>dwobject</i>	String specifying the DataWindow object to be cached	<i>trans</i> (optional)	N_tr transaction object used to retrieve rows for <i>dwobject</i> If you do not specify this argument, <i>dwobject</i> must have been saved with the data	<i>initialload</i> (optional)	Boolean specifying whether to load the data immediately or wait until the first of_Refresh request
Argument	Description														
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)														
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)														
<i>id</i>	String identifying the cached DataWindow object														
<i>dwobject</i>	String specifying the DataWindow object to be cached														
<i>trans</i> (optional)	N_tr transaction object used to retrieve rows for <i>dwobject</i> If you do not specify this argument, <i>dwobject</i> must have been saved with the data														
<i>initialload</i> (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 <i>dwobject</i> is already registered, -1 if an error occurs, -2 if there is an error with <i>trans</i> , 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

Description	To cache data retrieved via a SQL statement Retrieves rows (using a specified SQL statement) and caches them.
Access	Public
Syntax	<code>application.instancename.of_Register (id, trans, sql {, initialload })</code>

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	String identifying the cached DataWindow object
<i>trans</i>	N_tr transaction object used to retrieve rows for <i>dwobject</i>

	Argument	Description
	<i>sql</i>	String specifying the SQL SELECT statement used to create the cache
	<i>initialload</i> (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 <i>id</i> 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

Description

Access

Syntax

To cache data from a DataWindow control

Caches rows from a passed DataWindow control.

Public

applicationinstancename.of_Register (id, dwcontrol {, initialload})

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

Description

To cache data from a DataStore

Caches rows from a passed DataStore.

Access

Public

Syntax

`applicationinstancename.of_Register (id, datastore {, initialload})`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	String identifying the cached DataWindow object
<i>datastore</i>	DataStore instance whose DataWindow object contains the data to be cached
<i>initialload</i> (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

`applicationinstancename.of_Register (id, dwobject, data)`

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

Description

To cache data from a file

Caches data from a file.

Access

Public

Syntax

applicationinstancename.of_Register (id, dwobject, filename {, initialload })

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	String identifying the cached object
<i>dwobject</i>	String specifying the DataWindow object to be cached
<i>filename</i>	String specifying the name of a tab-delimited file whose elements match <i>dwobject</i>
<i>initialload</i> (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

```
applicationinstancename.of_RegisterArgs ( id, arguments )
```

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i>	ID of the cached object
<i>arguments</i>	Twenty-element array of the Any data type specifying new arguments for the DataWindow object associated with <i>id</i>

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	<i>application.instancename.of_UnRegister ({ id })</i>
Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_dwcache (the n_cst_appmanager default for this is inv_dwcache)
<i>id</i> (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 <i>id</i> , the function unregisters all objects.
Examples	This example calls the of_UnRegister function:

```
gnv_app.inv_dwcache.of_UnRegister &
( "d_emplist" )
```

n_cst_dwsrv

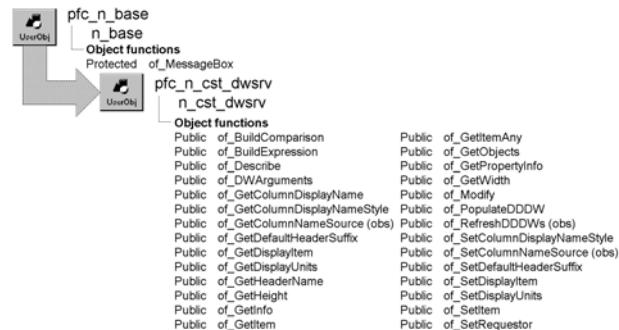
Description

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	<code>n_cst_dwsrv_dropdownsearch</code> <code>n_cst_dwsrv_filter</code> <code>n_cst_dwsrv_find</code> <code>n_cst_dwsrv_linkage</code> <code>n_cst_dwsrv_multitable</code> <code>n_cst_dwsrv_printpreview</code> <code>n_cst_dwsrv_property</code> <code>n_cst_dwsrv_querymode</code> <code>n_cst_dwsrv_report</code> <code>n_cst_dwsrv_reqcolumn</code> <code>n_cst_dwsrv_resize</code> <code>n_cst_dwsrv_rowmanager</code> <code>n_cst_dwsrv_rowselection</code> <code>n_cst_dwsrv_sort</code>
See also	<code>u_dw</code> <code>n_cst_dssrv</code>

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 <code>n_cst_dwsrv</code> instance	<code>u_dw</code>	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_GetItemAny
of_BuildExpression	of_GetObjects
of_Describe	ofGetPropertyInfo
of_DWArguments	ofGetWidth
of_GetColumnDisplayName	of_Modify
of_GetColumnDisplayNameStyle	of_PopulateDDDW
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_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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>row</i>	Long specifying the row number containing the column value
<i>column</i>	String specifying the name of the column containing the value
<i>value</i> (optional)	String specifying a value to be cast to the Any datatype. If you specify this argument, the function does not use <i>row</i> and <i>column</i>

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>row</i>	Long specifying the row number containing the column value used in the comparison string
<i>column</i>	String specifying the column used in the comparison string
<i>operator</i> (optional)	String specifying the operator used in the comparison string (the default is =)
<i>value</i> (optional)	String specifying a value used in the comparison string. If you specify this argument, the function does not use <i>row</i> and <i>column</i> 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

Description Describes the specified attribute for multiple columns or bands.

Access Public

Syntax dwcontrol.*instancename*.**of_Describe** (*returnndescribe*, *attribute* {, *objecttype*, *band*, *visibleonly* })

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>returnndescribe</i>	N_cst_dwobjectattrib instance into which the function places the described attributes
<i>attribute</i>	String specifying the name of the attribute to describe
<i>objecttype</i> (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 680

Argument	Description
<i>band</i> (optional)	String specifying the band to describe (specify * to return all bands). If you specify <i>objecttype</i> , you must specify this argument. The default is *, which describes all bands. Valid bands are: <ul style="list-style-type: none"> • Header • Detail • Footer • Summary • Header.<i>levelnumber</i> • Trailer.<i>levelnumber</i>
<i>visibleonly</i> (optional)	Boolean indicating whether to describe visible columns only (TRUE) or all columns (FALSE). If you specify <i>objecttype</i> , 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

dwcontrolinstancename.of_Describe (returndescribe, attribute, columnname)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>returndescribe</i>	N_cst_dwobjectattrib instance into which the function places the described attributes
<i>attribute</i>	String specifying the name of the attribute to describe
<i>columnname</i>	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.
Examples	Use Syntax 1 to describe multiple columns. 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 DropDownListDataWindow arguments. There are two syntaxes:

To	Use
Determine DropDownListDataWindow arguments	Syntax 1
Determine DataWindow arguments	Syntax 2

Syntax 1

To access DropDownListDataWindow arguments

Description	Determines if the passed dropdown DataWindow has retrieval arguments.
Access	Public

Syntax

```
dwcontrol.instancename.of_DWArguments ( dwchild, arguments,
datatypes )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>dwchild</i>	DataWindowChild variable specifying the DropDownListDataWindow to be tested for retrieval arguments
<i>arguments</i>	String array into which the function places argument names (passed by reference)
<i>datatypes</i>	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

dwcontrolinstancename.of_DWArguments (arguments, datatypes)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>arguments</i>	String array into which the function places argument names (passed by reference)
<i>datatypes</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>column</i>	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_SetColumnDisplayStyle 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_GetColumnDisplayStyle

Description Retrieves the DataWindow's column display name style.

Access Public

Syntax `dwcontrol.instancename.of_GetColumnDisplayStyle ()`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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_GetColumnDisplayStyle function:

```
Integer li_style

li_style = &
dw_1.inv_base.of_GetColumnDisplayStyle()
MessageBox("Style", &
"Display style is " + String(li_style))
```

of_GetColumnNameSource

Description Obsolete. Call of_GetColumnDisplayStyle instead.

Access Public

Syntax dwcontrol.*instancename*.**of_GetColumnNameSource** ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	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
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	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 <i>rows</i> , 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>columnname</i>	String specifying the DataWindow column name
<i>suffix</i> (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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>infoobject</i>	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

```
dwcontrolinstancename.of_GetItem ( row, column {, buffer, originalvalue } )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>row</i>	Long specifying the row containing the column to be returned
<i>column</i>	String or integer identifying the column to be returned
<i>buffer</i> (optional)	DWBuffer enumerated data type optionally specifying the DataWindow buffer from which to return the value. The default is Primary!
<i>originalvalue</i> (optional)	Boolean specifying whether to return the original value (TRUE) or the current DataWindow value (FALSE). The default is FALSE. IF you specify <i>buffer</i> , 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>row</i>	Long specifying the row number of the value to be retrieved
<i>column</i>	String or integer specifying the name of the column for the value to be retrieved
<i>buffer</i> (optional)	String specifying the name of the DataWindow buffer from which to get the column's data value. The default is Primary!
<i>originalvalue</i> (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	<code>dwcontrolinstancename.of_GetObjects (objectlist {, objecttype, band, visibleonly })</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>objectlist</i>	String array into which the function places object names (passed by reference)
<i>objecttype</i> (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
<i>band</i> (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
<i>visibleonly</i> (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 <i>type</i> , <i>band</i> , and <i>visibleonly</i> . Specifying an asterisk (*) for either <i>objecttype</i> or <i>band</i> means that you want all occurrences of objects across type or band. Specifying TRUE for <i>visibleonly</i> 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	<code>dwcontrol.instancename.ofGetPropertyInfo (propertyobject)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv (the u_dw default is inv_base)</td></tr> <tr> <td><i>propertyobject</i></td><td>n_cst_propertyattrib instance into which the function places property information (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)	<i>propertyobject</i>	n_cst_propertyattrib instance into which the function places property information (passed by reference)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)								
<i>propertyobject</i>	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 ofGetPropertyInfo function:								

```

n_cst_propertyattrib  lnv_prop

dw_1.inv_base.ofGetPropertyInfo(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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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 ofGetWidth function to determine the width of the dw_employee DataWindow:

```
Long    ll_width

ll_width = dw_employee.inv_base.ofGetWidth( )
```

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

Modifies the specified attribute for multiple columns or bands.

Access Public

Syntax *dwcontrol.instancename.of_Modify (attribute, value
{, objecttype, band, visibleonly })*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>attribute</i>	String specifying the name of the attribute to be modified
<i>value</i>	String specifying the new value for the attribute

Argument	Description
<i>objecttype</i> (optional)	String specifying the type of objects to modify. The default is *, which modifies all types For more information on DataWindow types, see of_GetObjects on page 680
<i>band</i> (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
<i>visibleonly</i> (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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>attribute</i>	String specifying the name of the attribute to be modified
<i>value</i>	String specifying the new value for the attribute
<i>columnname</i>	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 or columns.

Access Public

Syntax *dwcontrol.instanceName.of_PopulateDDDw ({column})*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instanceName</i>	Instance name of <i>n_cst_dwsrv</i> (the u_dw default is <i>inv_base</i>)
<i>column</i> (optional)	String specifying the column name or integer specifying the column number. If you omit this argument, the function populates all DropDownDataWindows in a DataWindow

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

Usage This function calls the u_dw pfc_PopulateDDDw event. You must extend this event to call the PowerScript Retrieve function for the passed DropDownDataWindow column, specifying retrieval arguments as necessary.

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_RefreshDDDWs

Description Obsolete. Call of_PopulateDDDW instead.

Access Public

Syntax dwcontrol.*instancename*.**of_RefreshDDDWs** ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)

Return value Integer. Returns the number of DropDownListDataWindows found and refreshed.

of_SetColumnNameStyle

Description Specifies the DataWindow's column display name style.

Access Public

Syntax dwcontrol.*instancename*.**of_SetColumnNameStyle** (*style*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>style</i>	Integer or n_cst_dwsrv constant specifying the display style: <ul style="list-style-type: none"> • 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_SetColumnNameStyle function:

```
...
dw_1.inv_base.of_SetColumnNameStyle &
(dw_1.inv_base.HEADER)
```

of_SetColumnNameSource

Description Obsolete. Call of_SetColumnDisplayStyle instead.

Access Public

Syntax ***dwcontrol.instancename.of_SetColumnNameSource (source)***

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv</i> (the u_dw default is <i>inv_base</i>)
<i>source</i>	Integer specifying the column name source: <ul style="list-style-type: none"> • 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv</i> (the u_dw default is <i>inv_base</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv</i> (the u_dw default is <i>inv_base</i>)
<i>itemname</i>	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 <i>this row</i> , 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>units</i>	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 <i>rows</i> , 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>row</i>	Long specifying the row containing the column to be updated
<i>column</i>	String or integer identifying the column to be updated. The function automatically converts this into the appropriate data type
<i>value</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv (the u_dw default is inv_base)
<i>requestor</i>	U_dw-based DataWindow control to be associated with a service

Return value

None

Usage

The u_dw of_SetServiceName 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:

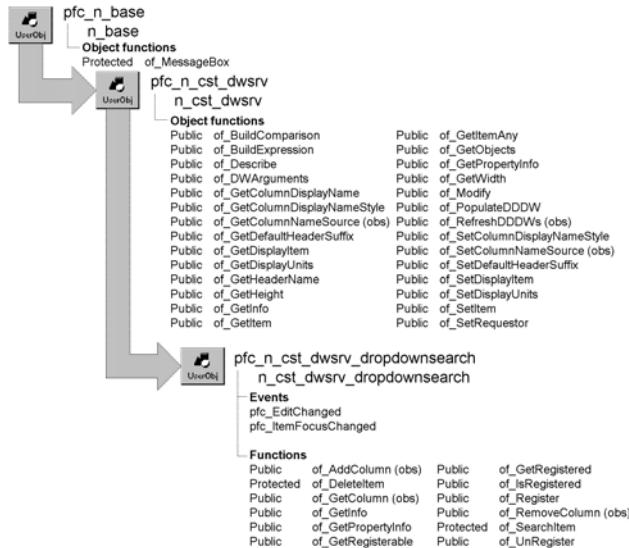
```

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)
```

PFC 7 custom class extension

To further the extendibility of PFC, the n_cst_dwsrv_dropdownsearchattrib custom class user object replaces the os_columns object structure that is now obsolete.

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
inv_columns []	Column and text information for dropdown DataWindows	n_cst_dwsrv_dropdownsearchattrib	Protected	Internal
is_textprev	Previously typed text	String	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>row</i>	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 <i>al_row</i> argument (passed by reference)
<i>dwobject</i>	DWOObject containing a reference to the column for the item whose value has been changed. <i>Dwobject</i> is a reference to the column object, not the name of the column. From within the pfc_EditChanged event, access this value through the <i>adwo_obj</i> argument (passed by reference)
<i>data</i>	String containing the new data the user has specified for the item. From within the pfc_EditChanged event, access this value through the <i>as_data</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>row</i>	Long containing the number of the row. From within the pfc_ItemFocusChanged event, access this value through the <i>al_row</i> argument

Argument	Description
<i>dwoobject</i>	DWObject containing a reference to the column containing the item whose value has been changed. <i>Dwoobject</i> is a reference to the column object, not the name of the column. From within the pfc_ItemFocusChanged event, access this value through the <i>adwo_object</i> 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 pre-coded object functions:

of_AddColumn	of_GetRegistered
of_DeleteItem	of_IsRegistered
of_GetColumn	of_Register
of_GetInfo	of_RemoveColumn
ofGetPropertyInfo	of_SearchItem
of_GetRegisterable	of_UnRegister

of_AddColumn

Description Obsolete. Call of_Register instead.

Access Public

Syntax dwcontrol.*instancename*.of_AddColumn ({ *column* })

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_dropdownsearch</i> (the u_dw default is <i>inv_dropdownsearch</i>)
<i>column</i> (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	<code>dwcontrol.instancename.of_DeleteItem (index)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>index</i>	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	<code>dwcontrol.instancename.of_GetColumn (columns)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>columns</i>	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 <i>columns</i> array.
--------------	--

of_GetInfo

Description	Retrieves dropdown search service information.
Access	Public
Syntax	<code>dwcontrolinstancename.of_GetInfo (infoobject)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>infoobject</i>	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)
```

ofGetPropertyInfo

Description	Retrieves information about the dropdown search service's properties.
Access	Public
Syntax	<code>dwcontrolinstancename.ofGetPropertyInfo (propertyobject)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>propertyobject</i>	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.ofGetPropertyInfo(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

dwcontrolinstancename.of_GetRegisterable (columns)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>columns</i>	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 DropDownListDataWindow 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_dropdownsearch</i> (the u_dw default is <i>inv_dropdownsearch</i>)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_dropdownsearch</i> (the u_dw default is <i>inv_dropdownsearch</i>)
<i>column</i>	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 DropDownListDataWindow and DropDownListBox columns with the dropdown search service.

Access Public

Syntax *dwcontrol.instancename.of_Register ({ column })*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<i>column</i> (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 DropDownListDataWindow 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	<code>dwcontrol.<i>instancename</i>.of_RemoveColumn (<i>column</i>)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_dropdownsearch</i> (the u_dw default is <i>inv_dropdownsearch</i>)
<i>column</i>	String specifying the dropdown DataWindow column to remove from the <i>istr_columns</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_dropdownsearch</i> (the u_dw default is <i>inv_dropdownsearch</i>)
<i>column</i>	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

`dwcontrolinstancename.of_UnRegister ({ column })`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of n_cst_dwsrv_dropdownsearch (the u_dw default is inv_dropdownsearch)
<code>column</code> (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:

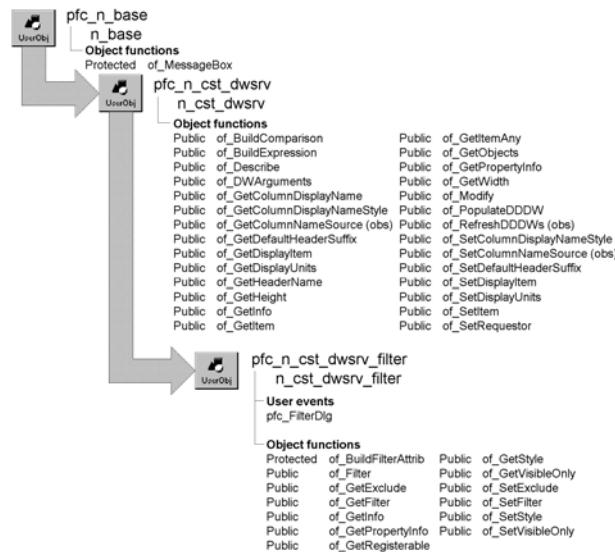
```
dw_1.inv_dropdownsearch.of_UnRegister( "dept_id" )
```

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 `_SetColumnDisplayStyle` function:

```
dw_emp.inv_filter.of_SetColumnDisplayStyle(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
<code>EXTENDED</code>	Constant set to 1	Integer	Public	Use with <code>of_SetStyle</code> to specify filter dialog box style
<code>ib_visibleonly</code>	Controls whether Filter dialog boxes display nonvisible columns	Boolean	Protected	Internal
<code>ii_style</code>	Tracks filter style	Integer	Protected	Internal
<code>is_excludedcolumns[]</code>	Tracks excluded columns	String	Protected	Internal
<code>is_filterin</code>	Filter the user specified	String	Protected	Internal
<code>is_filterout</code>	DataWindow filter	String	Protected	Internal
<code>SIMPLE</code>	Constant set to 2	Integer	Public	Use with <code>of_SetStyle</code> 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
ofGetPropertyInfo	of_SetVisibleOnly
of_GetRegisterable	

of_BuildFilterAttrib

Description Prepares the filter information required by the filter dialog boxes.

Access Protected

Syntax *dwcontrolinstancename.of_BuildFilterAttrib (filterattrib)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_filter</i> (the u_dw default is <i>inv_filter</i>)
<i>filterattrib</i>	<i>N_cst_filterattrib</i> 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

dwcontrolinstancename.of_Filter()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

dwcontrolinstancename.of_GetExclude(columns)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrol.instancename.of_GetInfo (infoobject)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
<i>infoobject</i>	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)
```

ofGetPropertyInfo

Description	Retrieves information about the filter service's properties.
Access	Public
Syntax	<code>dwcontrol.instancename.ofGetPropertyInfo (propertyobject)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
<i>propertyobject</i>	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.ofGetPropertyInfo(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.instancecname.of_GetRegisterable (columns)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancecname</i>	Instance name of <i>n_cst_dwsrv_filter</i> (the u_dw default is <i>inv_filter</i>)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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.instancecname.of_SetExclude (excludecols)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancecname</i>	Instance name of <i>n_cst_dwsrv_filter</i> (the u_dw default is <i>inv_filter</i>)
<i>excludecols</i>	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

dwcontrolinstancename.of_SetFilter (filter)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
<i>filter</i>	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

dwcontrolinstancename.of_SetStyle (filterstyle)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
<i>filterstyle</i>	Integer or n_cst_dwsrv_filter constant specifying the filter style: <ul style="list-style-type: none"> • 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)
gny_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	<code>dwcontrolinstancename.of_SetVisibleOnly (boolean)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_filter (the u_dw default is inv_filter)
<i>boolean</i>	Boolean indicating whether filter dialog boxes display visible columns only (TRUE) or all columns (FALSE)

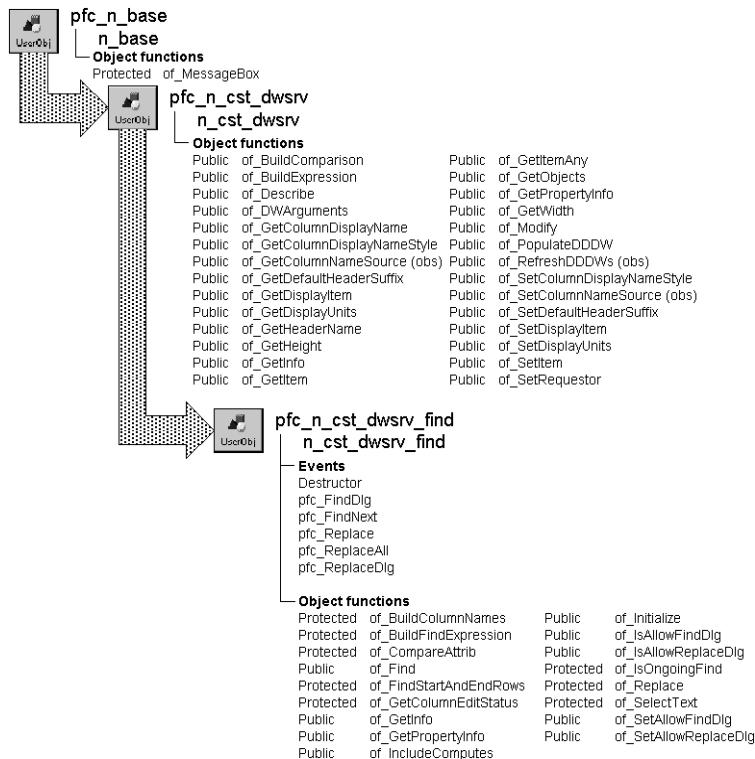
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Examples	This example calls the of_SetVisibleOnly function:

```
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_includecomputes	Indicates whether a find operation includes computed fields	Boolean	Protected	Use of_IncludeComputes 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

Instance variable	Description	Data type	Access	Usage
inv_findattrib	Structure object containing find and replace information	n_cst_findattrib	Protected	Internal
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. To include computed fields in a search, call of_IncludeComputes (true) before this event is triggered.

pfc_FindNext

Description	Continues a search or finds the next occurrence.
Syntax	<i>instancename</i> .EVENT pfc_FindNext (<i>findattrib</i>)
Argument	Description
<i>instancename</i>	Instance name of n_cst_dwsrv_find
<i>findattrib</i>	n_cst_findattrib instance containing find information. From within the pfc_FindNext event, access this value through the <i>anv_findattrib</i> 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
<i>instancename</i>	Instance name of n_cst_dwsrv_find
<i>findattrib</i>	n_cst_findattrib instance containing find information. From within the pfc_Replace event, access this value through the <i>anv_findattrib</i> argument

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

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
<i>instancename</i>	Instance name of n_cst_dwsrv_find
<i>findattrib</i>	n_cst_findattrib instance containing find information. From within the pfc_ReplaceAll event, access this value through the <i>any_findattrib</i> argument

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

The **w_replace** dialog box calls this event when the user clicks Replace All.

pfc_ReplaceDlq

Description Displays the w_replace dialog box.

This event is called when the DataWindow has focus and the user selects Edit>Replace from a menu that descends from the PFC 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
ofGetPropertyInfo	of_SetAllowReplaceDlg
of_IncludeComputes	

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
<i>replacelist</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_find</i> (the u_dw default is <i>inv_find</i>)
<i>find</i>	String being searched for
<i>column</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_find</i> (the u_dw default is <i>inv_find</i>)
<i>findattrib</i>	<i>n_cst_findattrib</i> 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

dwcontrolinstancename.of_Find ({ column, findstring, startrow, endrow })

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
<i>column</i> (optional)	String specifying the column to be searched
<i>findstring</i> (optional)	String to be searched for
<i>startrow</i> (optional)	Long specifying the row at which to start the search
<i>endrow</i> (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

dwcontrolinstancename.of_FindStartAndEndRows (startrow, endrow)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_find</i> (the u_dw default is <i>inv_find</i>)
<i>startrow</i>	Long into which the function places the start row (passed by reference)
<i>endrow</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
<i>column</i>	String specifying the column to test
<i>row</i>	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_currow)

//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.instancecname.of_GetInfo (infoobject)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancecname</i>	Instance name of <i>n_cst_dwsrv_find</i> (the u_dw default is <i>inv_find</i>)
<i>infoobject</i>	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)
```

ofGetPropertyInfo

Description

Retrieves information about the find service's properties.

Access

Public

Syntax	<code>dwcontrolinstancename.of_GetPropertyInfo (propertyobject)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
<i>propertyobject</i>	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:
	<pre>n_cst_propertyattrib lnv_prop dw_1.inv_find.ofGetPropertyInfo(lnv_prop) MessageBox("Info", & "Description: " + lnv_prop.is_description & + ". Name: " + lnv_prop.is_name & + ". Property tab text: " + & lnv_prop.is_propertytabtext)</pre>

of_IncludeComputes

Description	Determines whether computed fields are included in a search that uses the pfc_FindDlg event.
Access	Public
Syntax	<code>instancename.of_IncludeComputes (ab_switch)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_dwsrv_find
<i>ab_switch</i>	Boolean. If value is TRUE, computed fields will be included in subsequent searches
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Usage	Call this function before the pfc_FindDlg event is triggered.
Examples	This example calls the of_IncludeComputes function:
	<code>dw_emp.inv_find.of_IncludeComputes (true)</code>

of_Initialize

Description Initializes the `inv_findattrib` instance with the passed values.

Access Public

Argument	Description
<i>instancename</i>	Instance name of n_cst_dwsrv_find
<i>findattrib</i>	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.

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 dwcontrolinstancename.of_IsAllowFindDlg ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

n_cst_dwsrv_find

Syntax	<code>dwcontrolinstancename.of_IsOngoingFind (findattrib)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_find
<i>findattrib</i>	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: <pre>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 ...</pre>

of_Replace

Description	Replaces the specified row/column with the specified text.
Access	Protected
Syntax	<code>dwcontrolinstancename.of_Replace (row, column, replacewith)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
<i>row</i>	Long specifying the row containing the column whose text is to be replaced
<i>column</i>	String specifying the column whose text is to be replaced
<i>replacewith</i>	String specifying the new text

Return value	Integer. Returns 1 if the function succeeds, -2 if <i>replacewith</i> is not a valid value for <i>column</i> , 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
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of n_cst_dwsrv_find (the u_dw default is inv_find)
<code>column</code>	String specifying the column name
<code>row</code>	Long specifying the row number
<code>matchcase</code>	Boolean indicating whether the search is case-sensitive
<code>find</code>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_find</i> (the u_dw default is <i>inv_find</i>)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_find</i> (the u_dw default is <i>inv_find</i>)
<i>boolean</i>	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)

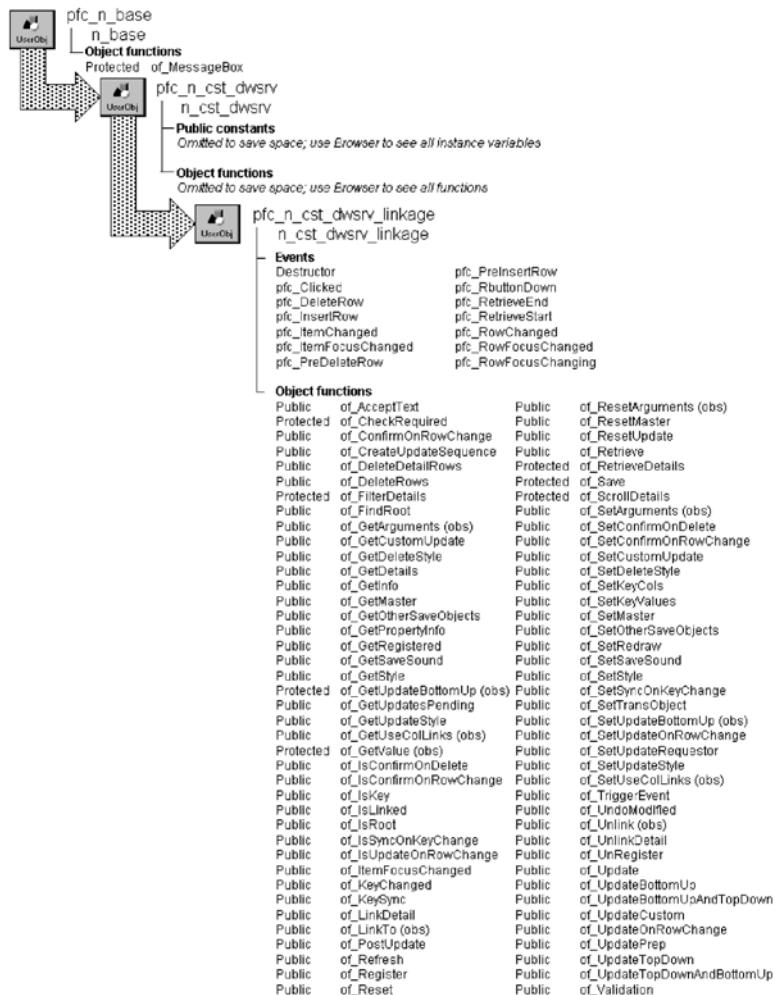
Confirm on row change When the master changes rows, this option displays a confirmation dialog box if modifications made to detail DataWindows will be lost (retrieval style only)

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 completely integrated with n_cst_luw and with the w_master pfc_Save process.

Ancestry



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

If you share data, don't add rows in the detail

You can use the PowerScript ShareData function to share data between master and detail DataWindows. However, do not insert rows into the detail when sharing data.,.

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_collapseuse	Indicates how PFC uses column links to refresh data: <ul style="list-style-type: none"> • 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
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

Instance variable	Description	Data type	Access	Usage
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*, *dwoobject*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_Clicked event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_Clicked event, access this value through the <i>ai_ypos</i> argument
<i>row</i>	Long indicating the clicked row. From within the pfc_Clicked event, access this value through the <i>al_row</i> argument
<i>dwoobject</i>	DWOObject variable indicating the clicked object. This argument is passed by reference. From within the pfc_Clicked event, access this value through the <i>adwo_obj</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long specifying the row that was deleted. From within the pfc_DeleteRow event, access this value through the <i>al_row</i> 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 *dwcontrolinstancename.Event pfc_InsertRow (row)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long specifying the row that was inserted. From within the pfc_InsertRow event, access this value through the <i>al_row</i> 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 *dwcontrolinstancename.EVENT pfc_ItemChanged (row, dwobject, data)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long indicating the number of the row in which the ItemChanged event occurred. From within the pfc_ItemChanged event, access this variable through the <i>al_row</i> argument
<i>dwobject</i>	DWOBJET variable indicating the changed item. From within the pfc_ItemChanged event, access this value through the <i>adwo_obj</i> argument
<i>data</i>	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 *dwcontrolinstancename.EVENT pfc_ItemFocusChanged (row, dwobject)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long indicating the number of the row in which the ItemFocusChanged event occurred. From within the pfc_ItemFocusChanged event, access this variable through the <i>al_row</i> argument
<i>dwoobject</i>	DWOObject variable indicating the changed item. From within the pfc_ItemFocusChanged event, access this value through the <i>adwo_obj</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)

n_cst_dwsrv_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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_RButtonDown event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_RButtonDown event, access this value through the <i>ai_ypos</i> argument
<i>row</i>	Long indicating the number of the row in which the RButtonDown event occurred. From within the pfc_RButtonDown event, access this variable through the <i>al_row</i> argument
<i>dwobject</i>	DWObject variable indicating the clicked item. From within the pfc_RButtonDown event, access this value through the <i>adwo_obj</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>rowcount</i>	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	<code>dwcontrolinstancename.EVENT pfc_RetrieveStart ()</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrolinstancename.EVENT pfc_RowChanged ()</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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.
-------------	---

n_cst_dwsrv_linkage

Syntax	<code>dwcontrol.instancename.Event pfc_RowFocusChanged (row)</code>								
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)</td></tr><tr><td><i>row</i></td><td>Long specifying the new row. From within the pfc_RowFocusChanged event, access this value through the <i>al_row</i> argument</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)	<i>row</i>	Long specifying the new row. From within the pfc_RowFocusChanged event, access this value through the <i>al_row</i> argument
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)								
<i>row</i>	Long specifying the new row. From within the pfc_RowFocusChanged event, access this value through the <i>al_row</i> 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 <code>o_SetUpdateOnRowChange</code> function.										
Syntax	<code>dwcontrol.instancename.EVENT pfc_RowFocusChanging (previousrow, newrow)</code>										
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)</td></tr><tr><td><i>previousrow</i></td><td>Long specifying the previous row. From within the pfc_RowFocusChanging event, access this value through the <i>al_prevrow</i> argument</td></tr><tr><td><i>newrow</i></td><td>Long specifying the new row. From within the pfc_RowFocusChanging event, access this value through the <i>al_newrow</i> argument</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)	<i>previousrow</i>	Long specifying the previous row. From within the pfc_RowFocusChanging event, access this value through the <i>al_prevrow</i> argument	<i>newrow</i>	Long specifying the new row. From within the pfc_RowFocusChanging event, access this value through the <i>al_newrow</i> argument
Argument	Description										
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control										
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)										
<i>previousrow</i>	Long specifying the previous row. From within the pfc_RowFocusChanging event, access this value through the <i>al_prevrow</i> argument										
<i>newrow</i>	Long specifying the new row. From within the pfc_RowFocusChanging event, access this value through the <i>al_newrow</i> 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 pre-coded object functions:

<code>of_AcceptText</code>	<code>of_ResetArguments</code>
<code>of_CheckRequired</code>	<code>of_ResetMaster</code>

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
ofGetPropertyInfo	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_UndoModified
of_IsRoot	of_Unlink
of_IsSyncOnKeyChange	of_UnlinkDetail
of_IsUpdateOnRowChange	of_UnRegister
of_ItemFocusChanged	of_Update
of_KeyChanged	of_UpdateBottomUp
of_KeySync	of_UpdateBottomUpAndTopDown
of_LinkDetail	of_UpdateCustom
of_LinkTo	of_UpdateOnRowChange
of_PostUpdate	of_UpdatePrep
of_Refresh	of_UpdateTopDown
of_Register	of_UpdateTopDownAndBottomUp
of_Reset	of_Validation

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>focusonerror</i> (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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>buffer</i>	DWBuffer variable indicating the DataWindow buffer to check for required fields

Argument	Description
<i>row</i>	Long specifying the number of the row to check. This argument also returns the number of the row in error (passed by reference)
<i>column</i>	Integer specifying the number of the first column to check. This argument also returns the number of the column in error (passed by reference)
<i>colname</i>	String into which the function places the name of the column in error (passed by reference)
<i>updateonly</i>	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_validatorow
    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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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 cause 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

dwcontrolinstancename.of_ConfirmOnRowChange()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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

dwcontrolinstancename.of_CreateUpdateSequence(sequence)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

Argument	Description
<i>sequence</i>	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:
	<pre> 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 </pre>

of_DeleteDetailRows

Description	Deletes detail rows when the corresponding master row is deleted.
Access	Public
Syntax	<code>dwcontrolinstancename.of_DeleteDetailRows ()</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control

Argument	Description
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>delete</i>	Boolean whether to delete the rows (TRUE) or discard them (FALSE)
<i>masterkeys</i>	String array containing column names from the deleted master row
<i>instancename</i>	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	<code>dwcontrol.instancename.of_FilterDetails (row)</code>								
	<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th><th style="text-align: center;">Description</th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td style="text-align: center;"><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)</td></tr> <tr> <td style="text-align: center;"><i>row</i></td><td>Long specifying the current row of the master DataWindow control</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)	<i>row</i>	Long specifying the current row of the master DataWindow control
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)								
<i>row</i>	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	<code>dwcontrol.instancename.of_FindRoot (root)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control

Argument	Description
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>root</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>linkageinfo</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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 dwcontrolinstancename.of_GetDeleteStyle()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrolinstancename.of_GetDetails (details)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default for this value is inv_linkage)</td></tr> <tr> <td><i>details</i></td><td>U_dw array into which the function places references to all details of <i>dwcontrol</i> (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this value is inv_linkage)	<i>details</i>	U_dw array into which the function places references to all details of <i>dwcontrol</i> (passed by reference)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this value is inv_linkage)								
<i>details</i>	U_dw array into which the function places references to all details of <i>dwcontrol</i> (passed by reference)								
Return value	Integer. Returns the number of elements in the <i>details</i> array.								
Usage	Call this function from any DataWindow in the linkage chain.								
	When using the of_Unlink function along with this function, check that all the elements in the Linkage chain reference a u_dw object as you loop through the linkage chain. Note that the of_Unlink function does not remove elements from the PFC Linkage chain. It only changes the element references to point to a random section in memory.								
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	<code>dwcontrolinstancename.of_GetInfo (infoobject)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)</td></tr> <tr> <td><i>infoobject</i></td><td>n_cst_infoattrib instance into which the function places service information (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)	<i>infoobject</i>	n_cst_infoattrib instance into which the function places service information (passed by reference)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)								
<i>infoobject</i>	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

dwcontrolinstancename.of_GetMaster (master)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>master</i>	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

dwcontrolinstancename.of_GetOtherSaveObjects (objects)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>objects</i>	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

```

ofGetPropertyInfo

Description

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrol.instancename.ofGetPropertyInfo (propertyobject)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)
<i>propertyobject</i>	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 ofGetPropertyInfo function:

```

n_cst_propertyattrib lnv_prop

dw_1.inv_linkage.ofGetPropertyInfo(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	<i>dwcontrolinstancename.of_GetRegistered (linkattrib)</i>								
	<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th><th style="text-align: center;">Description</th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td style="text-align: center;"><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)</td></tr> <tr> <td style="text-align: center;"><i>linkattrib</i></td><td>N_cst_linkageattrib instance into which the function places linkage argument values (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)	<i>linkattrib</i>	N_cst_linkageattrib instance into which the function places linkage argument values (passed by reference)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default for this is inv_linkage)								
<i>linkattrib</i>	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_SaveSound

Description	Retrieves the name of the sound file, if enabled, that is played after a successful save.
Access	Public
Syntax	<i>dwcontrolinstancename.of_SaveSound ()</i>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

dwcontrolinstancename.of_GetValue (row, column)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	String specifying the row containing the item to be accessed
<i>column</i>	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

dwcontrolinstancename.of_IsConfirmonDelete ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>column</i>	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

dwcontrolinstancename.of_IsLinked ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

dwcontrolinstancename.of_IsRoot ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

dwcontrolinstancename.of_IsSyncOnKeyChange ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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

`dwcontrolinstancename.of_IsUpdateOnRowChange ()`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

`dwcontrolinstancename.of_ItemFocusChanged (row, column)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long specifying the row with focus
<i>column</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long specifying the row number of the changed column in the master DataWindow
<i>column</i>	String specifying the key column in the master DataWindow
<i>prevvalue</i>	String containing the previous value
<i>newvalue</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	Long specifying the row number of the changed column in the master DataWindow
<i>column</i>	String specifying the key column in the master DataWindow
<i>prevvalue</i>	String containing the previous value
<i>newvalue</i>	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

`dwcontrolinstancename.of_LinkDetail (detail)`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<code>detail</code>	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	<code>dwcontrol.instancename.of_LinkTo (master)</code>								
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)</td></tr><tr><td><i>master</i></td><td>DataWindow variable pointing to the master DataWindow</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)	<i>master</i>	DataWindow variable pointing to the master DataWindow
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)								
<i>master</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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

`dwcontrolinstancename.of_Refresh (row)`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<code>row</code>	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

`dwcontrolinstancename.of_Register (mastercol, detailcol)`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<code>mastercol</code>	String specifying the column in the master DataWindow that is a linkage argument
<code>detailcol</code>	String specifying the column in the detail DataWindow that relates to <code>mastercol</code>

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 *dwcontrolinstancename.of_Reset ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>).

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 *dwcontrolinstancename.of_ResetArguments ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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 *dwcontrolinstancename.of_ResetMaster ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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.instancecname.of_Retrieve ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancecname</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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`dwcontrolinstancename.of_RetrieveDetails (row)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	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`dwcontrolinstancename.of_Save (validerow, prompt)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>validerow</i>	Long specifying the row to validate. Specify 0 if there is no row to validate
<i>prompt</i>	Boolean specifying whether the user should be prompted to save changes (TRUE) or changes should be saved automatically (FALSE)

Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none">• 2 Canceled• 1 Successful• 0 Updates pending• -1 An error occurred
Usage	<p>This function saves data by calling the associated window's pfc_SaveObjects function. It is called by <i>n_cst_dwsrv_linkage</i> events when update on row change is enabled.</p>
Examples	<p>This example is from the pfc_RowChanged event:</p>

```
...
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	<i>dwcontrolinstancename.of_ScrollDetails (row)</i>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>row</i>	Long specifying the current row in the master DataWindow

Return value	<p>Integer. Returns 1 if the function succeeds and -1 if an error occurs.</p>
Usage	<p>Internal.</p>

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

dwcontrolinstancename.of_SetArguments (mastercol, detailcol)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>mastercol</i>	String specifying the column in the master DataWindow that is a linkage argument
<i>detailcol</i>	String specifying the column in the detail DataWindow that relates to <i>mastercol</i>

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

dwcontrolinstancename.of_SetConfirmonDelete (boolean)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>boolean</i>	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	<code>dwcontrolinstancename.of_SetConfirmOnRowChange (boolean)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>boolean</i>	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	<code>dwcontrol.instancename.of_SetCustomUpdate (id)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)</td></tr> <tr> <td><i>id</i></td><td>Integer from 1 to 9 specifying the custom update identifier</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)	<i>id</i>	Integer from 1 to 9 specifying the custom update identifier
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)								
<i>id</i>	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:								
	<code>dw_cust.inv_linkage.of_SetCustomUpdate(9)</code>								

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	<code>dwcontrol.instancename.of_SetDeleteStyle (style)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)</td></tr> <tr> <td><i>style</i></td><td> Integer specifying the delete style: <ul style="list-style-type: none"> • 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 </td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)	<i>style</i>	Integer specifying the delete style: <ul style="list-style-type: none"> • 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
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)								
<i>style</i>	Integer specifying the delete style: <ul style="list-style-type: none"> • 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>column</i>	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

```
dwcontrolinstancename.of_SetKeyValues ( row )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>row</i>	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

```
dwcontrolinstancename.of_SetMaster ( masterdw )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>masterdw</i>	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	<code>dwcontrol.instancename.of_SetOtherSaveObjects (objects)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>objects</i>	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 <code>of_SetUpdateOnRowChange(TRUE)</code> .
Examples	This example calls the <code>of_SetOtherSaveObjects</code> 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	<code>dwcontrol.instancename.of_SetRedraw (boolean)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>boolean</i>	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 <code>of_SetRedraw</code> 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	<code>dwcontrol.instancename.of_SetSaveSound (soundfile)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>soundfile</i>	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 <i>soundfile</i> to an empty string. This feature is only used with the retrieval style.
Examples	This example calls the of_SetSaveSound function:
	<code>dw_cust.inv_linkage.of_SetSaveSound("slurp.wav")</code>

of_SetStyle

Description	Specifies how the linkage service refreshes detail DataWindows:
	<ul style="list-style-type: none"> • 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	<code>dwcontrol.instancename.of_SetStyle (style)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>style</i>	Integer or n_cst_dwsrv_linkage constant specifying the style: <ul style="list-style-type: none"> • 1 or 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	<code>dwcontrol.instancename.of_SetSyncOnKeyChange (boolean)</code>

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<code>boolean</code>	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

`dwcontrolinstancename.of_SetTransObject (transaction)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>transaction</i>	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

`dwcontrolinstancename.of_SetUpdateBottomUp (direction)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>direction</i>	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 *dwcontrolinstancename.of_SetUpdateOnRowChange ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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 *dwcontrolinstancename.of_SetUpdateRequestor (requestor)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)
<i>requestor</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>updatestyle</i>	Integer or n_cst_dwsrv_linkage constant specifying the update style: <ul style="list-style-type: none"> • 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>linktype</i>	Integer specifying the linkage type: <ul style="list-style-type: none">• 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>userevent</i>	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_UndoModified

Description Undoes pending changes to the requesting DataWindow or to all the DataWindows in the linkage chain.

Access Public

Syntax ***dwcontrol.instancename.of_UndoModified ({ab_all})***

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>ab_all</i>	Boolean to indicate whether or not the changes should be undone in the entire linkage chain or only in the requesting DataWindow: TRUE—Changes undone in entire linkage chain FALSE—Changes undone in requesting DataWindow

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

Usage You do not need to enter an argument to restrict the Undo operation to the requesting DataWindow.

Examples This example undoes pending changes in all the DataWindows in the linkage chain:

```
dw_master.inv_linkage.of_UndoModified (TRUE)
```

of_Unlink

Description Obsolete. Call of_ResetMaster instead.

Access Public

Syntax ***dwcontrol.instancename.of_Unlink ()***

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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.

n_cst_dwsrv_linkage

Access

Public

Syntax

dwcontrolinstancename.of_UnlinkDetail (detaildatawindow)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>detaildatawindow</i>	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

dwcontrolinstancename.of_UnRegister ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)
<i>accepttext</i> (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
<i>resetflags</i> (optional)	Boolean indicating whether <i>dwcontrol</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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 function that contains custom update processing.								
Access	Public								
Syntax	<code>dwcontrolinstancename.of_UpdateCustom (id)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)</td></tr> <tr> <td><i>id</i></td><td>Integer identifying the custom update, as specified in a previously called of_SetCustomUpdate function</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)	<i>id</i>	Integer identifying the custom update, as specified in a previously called of_SetCustomUpdate function
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_linkage (the u_dw default is inv_linkage)								
<i>id</i>	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	<p>The of_Update function calls this function when you set the update style to Custom via the of_SetUpdateStyle function.</p> <p>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.</p>								

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

dwcontrolinstancename.of_UpdateOnRowChange ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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

`dwcontrolinstancename.of_UpdatePrep()`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	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

`dwcontrolinstancename.of_UpdateTopDown()`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	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

dwcontrolinstancename.of_UpdateTopDownAndBottomUp ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_linkage</i> (the u_dw default is <i>inv_linkage</i>)

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

`dwcontrolinstancename.of_Validation ()`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

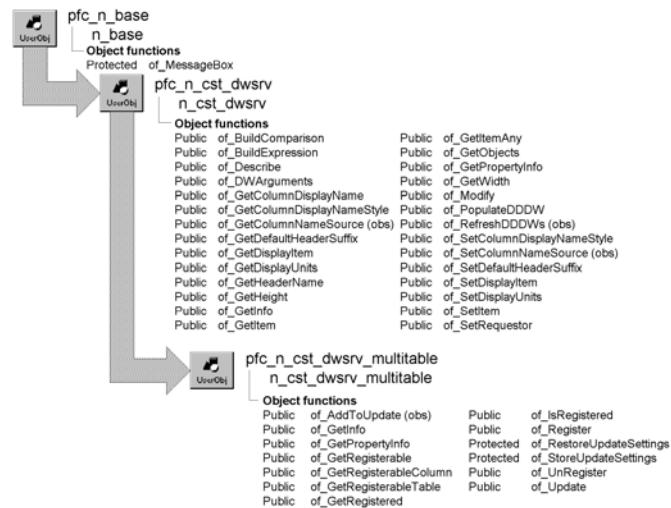
Description

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( )
```

PFC 7 custom class extension

To further the extendibility of PFC, a new n_cst_dwsrv_multitableattrib custom class user object (n_cst_dssrv_multitableattrib for n_cst_dssrv_multitable) replaces the os_updcharacteristics object structure that is now obsolete.

See also

- n_cst_dwsrv_rowmanager
- n_cst_dwsrv_linkage
- u_dw

Instance variables

N_cst_dwsrv_multitable includes the following instance variable:

Instance variable	Description	Data type	Access	Usage
inv_newupdate [] (replaces istr_newupdate[])	Used to control multiple updates	n_cst_dwsrv_multitableattrib (n_cst_dssrv_multitableattrib for n_cst_dssrv_multitable)	Protected	Internal

Functions

N_cst_dwsrv_multitable includes precoded object functions:

of_AddToUpdate	of_IsRegistered
of_GetInfo	of_Register
ofGetPropertyInfo	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>table</i>	String specifying the table name
<i>keycolumns</i>	String array containing the key columns used for update
<i>updateablecols</i> (optional)	String array containing the list of updatable columns (the default is all columns in the DataWindow)
<i>keyinplace</i> (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 <i>whereoption</i>
<i>whereoption</i> (optional)	Integer specifying the DataWindow Update Where option: <ul style="list-style-type: none"> • 0 Key columns • 1 Key and updatable columns • 2 Key and modified columns If you specify this argument, you must also specify <i>keyinplace</i>

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>infoobject</i>	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)
```

ofGetPropertyInfo

Description Retrieves information about the service's properties.

Access Public

Syntax dwcontrol.*instancename*.**ofGetPropertyInfo** (*propertyobject*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>propertyobject</i>	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

dwcontrolinstancename.of_GetRegisterable (columns)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>columns</i>	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

dwcontrolinstancename.of_GetRegisterableColumn (table, columns)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>table</i>	String specifying the name of the table
<i>columns</i>	String array into which the function places all registerable columns for <i>table</i> (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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>tables</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>table</i>	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	<code>dwcontrolinstancename.of_Register (table, keycolumns {, updatablecolumns {, keyinplace, whereoption } })</code>																
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of <i>n_cst_dwsrv_multitable</i> (the u_dw default is <i>inv_multitable</i>)</td></tr> <tr> <td><i>table</i></td><td>String specifying the table to be registered</td></tr> <tr> <td><i>keycolumns</i></td><td>String array specifying the key columns for <i>table</i></td></tr> <tr> <td><i>updatablecolumns</i> (optional)</td><td>String array specifying updatable columns for <i>table</i>. The default is all columns</td></tr> <tr> <td><i>keyinplace</i> (optional)</td><td>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 <i>whereoption</i></td></tr> <tr> <td><i>whereoption</i> (optional)</td><td> Integer specifying the WHERE option: <ul style="list-style-type: none"> • 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 <i>keyinplace</i> </td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of <i>n_cst_dwsrv_multitable</i> (the u_dw default is <i>inv_multitable</i>)	<i>table</i>	String specifying the table to be registered	<i>keycolumns</i>	String array specifying the key columns for <i>table</i>	<i>updatablecolumns</i> (optional)	String array specifying updatable columns for <i>table</i> . The default is all columns	<i>keyinplace</i> (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 <i>whereoption</i>	<i>whereoption</i> (optional)	Integer specifying the WHERE option: <ul style="list-style-type: none"> • 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 <i>keyinplace</i>
Argument	Description																
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control																
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_multitable</i> (the u_dw default is <i>inv_multitable</i>)																
<i>table</i>	String specifying the table to be registered																
<i>keycolumns</i>	String array specifying the key columns for <i>table</i>																
<i>updatablecolumns</i> (optional)	String array specifying updatable columns for <i>table</i> . The default is all columns																
<i>keyinplace</i> (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 <i>whereoption</i>																
<i>whereoption</i> (optional)	Integer specifying the WHERE option: <ul style="list-style-type: none"> • 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 <i>keyinplace</i>																

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>settings</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>settings</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>table</i> (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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_multitable (the u_dw default is inv_multitable)
<i>accept</i> (optional)	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database. The default is TRUE
<i>resetflag</i> (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

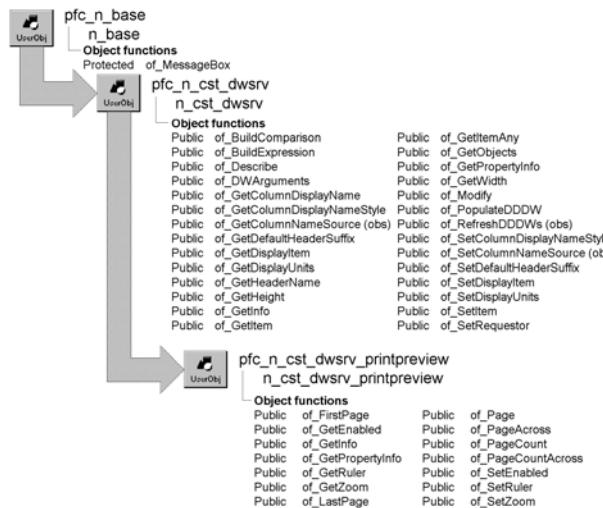
Description

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: <pre>dw_emp.of_SetPrintPreview(TRUE)</pre>
See also	2 Toggle print preview mode by calling the u_dw pfc_PrintPreview event: <pre>dw_emp.Event pfc_PrintPreview()</pre> 3 Call other print preview functions as needed. n_cst_dwsrv_report u_dw

Functions

`N_cst_dwsrv_printpreview` contains precoded object functions:

<code>of_FirstPage</code>	<code>of_Page</code>
<code>of_GetEnabled</code>	<code>of_PageAcross</code>
<code>of_GetInfo</code>	<code>of_PageCount</code>
<code>ofGetPropertyInfo</code>	<code>of_PageCountAcross</code>
<code>of_GetRuler</code>	<code>of_SetEnabled</code>
<code>of_GetZoom</code>	<code>of_SetRuler</code>
<code>of_LastPage</code>	<code>of_SetZoom</code>

of_FirstPage

Description	Scrolls to the first page.						
Access	Public						
Syntax	<code>dwcontrol.instancename.of_FirstPage ()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>dwcontrol</code></td> <td>Instance name of the u_dw-based DataWindow control</td> </tr> <tr> <td><code>instancename</code></td> <td>Instance name of <code>n_cst_dwsrv_printpreview</code> (the u_dw default is <code>inv_printpreview</code>)</td> </tr> </tbody> </table>	Argument	Description	<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control	<code>instancename</code>	Instance name of <code>n_cst_dwsrv_printpreview</code> (the u_dw default is <code>inv_printpreview</code>)
Argument	Description						
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control						
<code>instancename</code>	Instance name of <code>n_cst_dwsrv_printpreview</code> (the u_dw default is <code>inv_printpreview</code>)						
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.						
Usage	Not in <code>n_cst_dssrv_printpreview</code> This function is not implemented in <code>n_cst_dssrv_printpreview</code> .						

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
<i>infoobject</i>	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_preview.of_GetInfo(lnv_info)
MessageBox("Info", &
    "Description: " + lnv_info.is_description &
    ". Name: " + lnv_info.is_name)
```

ofGetPropertyInfo**Description**

Retrieves information about the service's properties.

Access

Public

Syntax

dwcontrolinstancename.ofGetPropertyInfo (propertyobject)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dssrv_preview (the u_dw default is inv_preview)
<i>propertyobject</i>	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_preview

This function is not implemented in n_cst_dssrv_preview.

Examples

This example calls the ofGetPropertyInfo function:

```
n_cst_propertyattrib lnv_prop

dw_1.inv_preview.ofGetPropertyInfo(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 *dwcontrolinstancename.of_GetRuler ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_printpreview</i> (the u_dw default is <i>inv_printpreview</i>)

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 *dwcontrolinstancename.of_GetZoom ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_printpreview</i> (the u_dw default is <i>inv_printpreview</i>)

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	<code>dwcontrolinstancename.of_LastPage()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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: <code>dw_emplist.inv_printpreview.of_LastPage()</code>

of_Page

Description	Retrieves the number of the current page.
Access	Public
Syntax	<code>dwcontrolinstancename.of_Page()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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: <code>Integer li_page li_page = dw_emp.inv_printpreview.of_Page()</code>

of_PageAcross

Description	Retrieves the number of the current horizontal page.
Access	Public

Syntax	<code>dwcontrol.instancename.of_PageAcross()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_printpreview</i> (the u_dw default is <i>inv_printpreview</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_printpreview</i> (the u_dw default is <i>inv_printpreview</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_printpreview (the u_dw default is inv_printpreview)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_printpreview</i> (the u_dw default is <i>inv_printpreview</i>)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_printpreview</i> (the u_dw default is <i>inv_printpreview</i>)
<i>zoompercentage</i> (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.

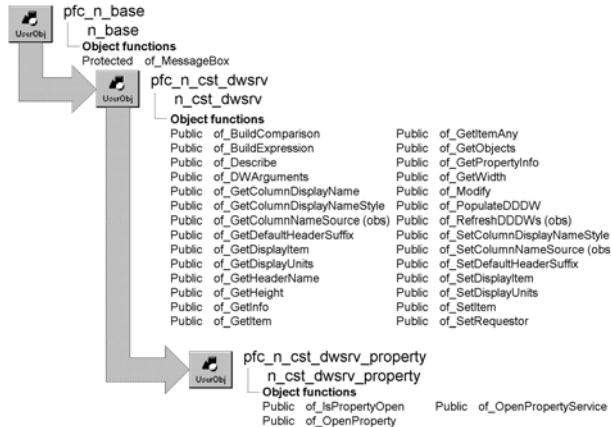
Examples This example calls the of_SetZoom function, displaying the w_zoom dialog box:

```
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
<code>iw_property</code>	Reference to <code>w_dwproperty</code> window	<code>w_dwproperty</code>	Protected	Internal
<code>iw_dwpropertyservices</code>	Reference to <code>w_dwpropertyservices</code> window	<code>w_dwpropertyservices</code>	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 *dwcontrolinstancename.of_IsPropertyOpen ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrolinstancename.of_OpenProperty (properties)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)</td></tr> <tr> <td><i>properties</i></td><td>N_cst_dwpropertyattrib instance containing properties to display</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)	<i>properties</i>	N_cst_dwpropertyattrib instance containing properties to display
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)								
<i>properties</i>	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	<code>dwcontrolinstancename.of_OpenPropertyService (service)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)</td></tr> <tr> <td><i>service</i></td><td>PowerObject containing the service to be displayed in the w_dwpropertyservice dialog box</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)	<i>service</i>	PowerObject containing the service to be displayed in the w_dwpropertyservice dialog box
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_property (the u_dw default is either inv_property or snv_property)								
<i>service</i>	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

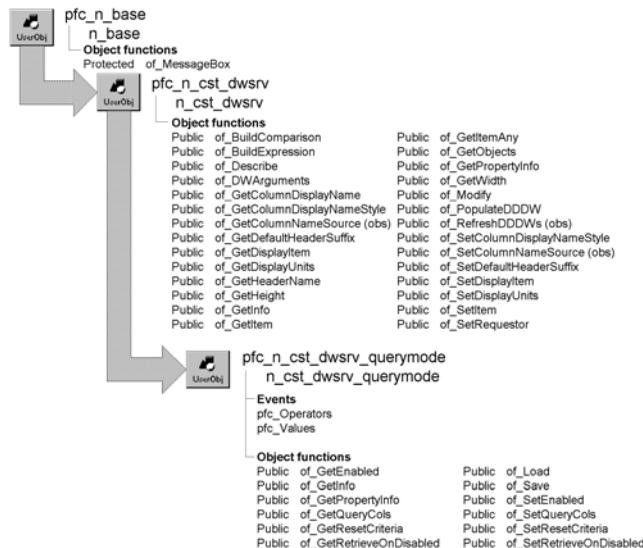
This example calls the of_OpenPropertyService function:

```
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.

PFC 7 custom class extension

To further the extendibility of PFC, a new *n_cst_dwsrv_querymodeattrib* custom class user object replaces the *os_querymodeinfo* object structure that is now obsolete.

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
<i>ib_resetquerycriteria</i>	Indicates whether PFC resets query criteria between query mode toggles	Boolean	Protected	Internal
<i>ib_retrieveondisabled</i>	Indicates whether PFC automatically retrieves the DataWindow when query mode is disabled	Boolean	Protected	Internal
<i>inv_querymodeinfo[]</i>	Stores querymode information for each of the DataWindow's columns	<i>n_cst_dwsrv_querymodeattrib</i>	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 <code>m_dw</code> 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 <code>m_dw</code> popup menu.

Functions

`N_cst_dwsrv_querymode` contains precoded object functions:

<code>of_GetEnabled</code>	<code>of_Load</code>
<code>of_GetInfo</code>	<code>of_Save</code>
<code>ofGetPropertyInfo</code>	<code>of_SetEnabled</code>
<code>of_GetQueryCols</code>	<code>of_SetQueryCols</code>
<code>of_GetResetCriteria</code>	<code>of_SetResetCriteria</code>
<code>of_GetRetrieveOnDisabled</code>	<code>of_SetRetrieveOnDisabled</code>

of_GetEnabled

Description Retrieves the DataWindow's current query mode state.

Access Public

Syntax dwcontrol.*instancename*.**of_GetEnabled** ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
<i>infoobject</i>	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

`dwcontrolinstancename.of_GetPropertyInfo (propertyobject)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
<i>propertyobject</i>	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 ofGetPropertyInfo 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

`dwcontrolinstancename.of_GetQueryCols (columns)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_querymode</i> (the u_dw default is <i>inv_querymode</i>)

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
	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
	<i>instancename</i>	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:
		<pre>Boolean lb_retrieve lb_retrieve = & dw_emp.inv_querymode.of_GetRetrieveOnDisabled()</pre>

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	<i>dwcontrol.instancename.of_Load (title, pathname, filename {, extension {, filter } })</i>

	Argument	Description
	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
	<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
	<i>title</i>	String specifying the title of the File Open dialog box
	<i>pathname</i>	String into which the function returns the path for the chosen file (passed by reference)
	<i>filename</i>	String into which the function returns the filename (passed by reference)
	<i>extension</i> (optional)	String specifying the 1- to 3-character default file extension. The default is TXT
	<i>filter</i> (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 `dwcontrolinstancename.of_Save (title, pathname, filename {, extension {, filter } })`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of <code>n_cst_dwsrv_querymode</code> (the u_dw default is <code>inv_querymode</code>)
<code>title</code>	String specifying the title of the File Save dialog box
<code>pathname</code>	String specifying the default path and filename. The function uses this argument to return the path for the saved file (passed by reference)
<code>filename</code>	String into which the function returns the filename (passed by reference)
<code>extension</code> (optional)	String specifying the 1- to 3-character default file extension. The default is <code>TXT</code>
<code>filter</code> (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, <code>*.*</code> or <code>*.exe</code>). The format for filter is: description, <code>*.ext</code> . 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 dwcontrolinstancename.of_SetEnabled (boolean)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
<i>boolean</i>	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

n_cst_dwsrv_querymode

Syntax	<code>dwcontrolinstancename.of_SetQueryCols (columns)</code>								
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)</td></tr><tr><td><i>columns</i></td><td>String array specifying the columns to be set as querymode columns</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)	<i>columns</i>	String array specifying the columns to be set as querymode columns
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)								
<i>columns</i>	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: <pre>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)</pre>								

of_SetResetCriteria

Description	Specifies whether PFC resets query criteria between query mode toggles.								
Access	Public								
Syntax	<code>dwcontrolinstancename.of_SetResetCriteria (boolean)</code>								
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)</td></tr><tr><td><i>boolean</i></td><td>Boolean indicating whether PFC resets query criteria (TRUE) or not (FALSE)</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)	<i>boolean</i>	Boolean indicating whether PFC resets query criteria (TRUE) or not (FALSE)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)								
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_querymode (the u_dw default is inv_querymode)
<i>boolean</i>	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

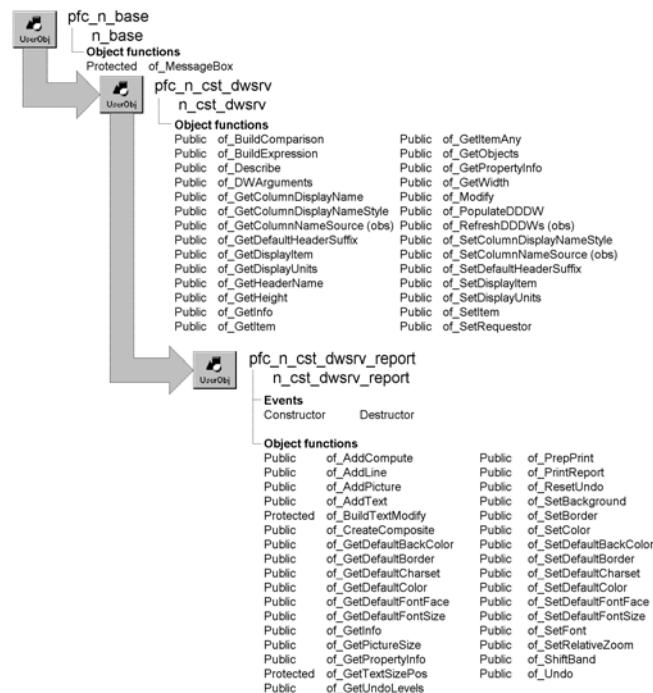
Description

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	<p>Use this service with report DataWindows to provide enhanced viewing and printing capabilities.</p> <p>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.</p> <p>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.</p> <p>To use this service:</p> <ol style="list-style-type: none"> 1 Enable the service using the u_dw of_SetReport function: <pre>dw_emp.of_SetReport(TRUE)</pre> <ol style="list-style-type: none"> 2 Call reporting and printing functions as needed.
See also	<p>n_cst_dwsrv_printpreview</p> <p>u_dw</p>

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
inv_platform	Reference variable for platform services	n_cst_platform	Protected	Internal

Instance variable	Description	Data type	Access	Usage
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
ofGetPropertyInfo	of_ShiftBand
of.GetTextSizePos	of_Undo
of_GetUndoLevels	

of_AddCompute**Description**

Adds a computed column to any band of a DataWindow.

Access

Public

Syntax

```
dwcontrolinstancename.of_AddCompute ( expression, band, xcoordinate,
ycoordinate {, border {, bold, italic, underline {, fontface, fontsize,
characterset {, color, backcolor {, execute } } } } } )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>expression</i>	String specifying the expression to be added as the computed column
<i>band</i>	String specifying the band to which the function will add the computed column. Valid bands are: Header Detail Footer Summary Header.levelnumber Trailer.levelnumber
<i>xcoordinate</i>	<p>Either of the following:</p> <ul style="list-style-type: none"> • Integer specifying the <i>x</i> location of the computed column. If you specify an integer for <i>xcoordinate</i>, you must also specify an integer for <i>ycoordinate</i> • Alignment enumerated data type (Center!, Justify!, Left!, or Right!). If you specify an alignment enumerated data type for <i>xcoordinate</i>, you must specify a VTextAlign enumerated data type for <i>ycoordinate</i>

Argument	Description
<i>ycoordinate</i> (optional)	<p>Either of the following:</p> <ul style="list-style-type: none"> • Integer specifying the <i>y</i> location of the computed column. If you specify an integer for <i>ycoordinate</i>, you must also specify an integer for <i>xcoordinate</i> • VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!). If you specify a VTextAlign enumerated data type for <i>ycoordinate</i>, you must specify an Alignment enumerated data type for <i>xcoordinate</i>
<i>border</i> (optional)	<p>Border enumerated data type specifying the border style:</p> <p>NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised!</p> <p>Specify the default border style using the of_SetDefaultBorder function</p>
<i>bold</i> (optional)	<p>Boolean specifying whether text is bold:</p> <p>TRUE—Bold FALSE—Not bold (default)</p>
<i>italic</i> (optional)	<p>Boolean specifying whether text is italic:</p> <p>TRUE—Italic FALSE—Not italic (default)</p>
<i>underline</i> (optional)	<p>Boolean specifying whether text is underlined:</p> <p>TRUE—Underlined FALSE—Not underlined (default)</p>
<i>fontface</i> (optional)	<p>String specifying the font to use</p> <p>Specify the default font face using the of_SetDefaultFontFace function</p>
<i>fontsize</i> (optional)	<p>Integer specifying the point size of the font for the computed column</p> <p>Specify the default font size using the of_SetDefaultFontSize function</p>

Argument	Description
<i>charset</i> (optional)	<p>FontCharSet enumerated data type specifying the character set to use for the computed column:</p> <ul style="list-style-type: none"> ANSI! Symbol! ChineseBig5! Hangul! ShiftJIS! OEM! <p>Specify the default character set using the <code>of_SetDefaultCharset</code> function</p>
<i>color</i> (optional)	<p>Long specifying the color of the computed field</p> <p>Specify the default color using the <code>of_SetDefaultColor</code> function</p>
<i>backcolor</i> (optional)	<p>Long specifying the background color of the computed field</p> <p>Specify the default background color using the <code>of_SetDefaultBackColor</code> function</p>
<i>execute</i> (optional)	<p>Boolean specifying whether to execute the <code>Modify</code> function that adds the computed column to the band (TRUE) or return a string that you use to execute the <code>Modify</code> function (FALSE). The default is TRUE</p>

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.

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

Description

Add a line by specifying the x and y coordinates of the endpoints

Adds a line by specifying the x and y coordinates of the endpoints.

Access

Public

Syntax

```
dwcontrolinstancename.of_AddLine ( band, penstyle, penwidth, x1, y1, x2,  
y2 {, color, backcolor {, execute } } )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)
<i>band</i>	String specifying the band to which the function will add the line. Valid bands are: Header Detail Footer Summary Header.levelnumber Trailer.levelnumber
<i>penstyle</i>	Linestyle enumerated data type specifying the pen style of the line: Continuous! Dash! Dot! DashDot! DashDotDot! Transparent!
<i>penwidth</i>	Integer specifying the pen width of the line
<i>x1</i>	Integer specifying the <i>x1</i> location of the line
<i>y1</i>	Integer specifying the <i>y1</i> coordinate of the line
<i>x2</i>	Integer specifying the <i>x2</i> location of the line
<i>y2</i>	Integer specifying the <i>y2</i> coordinate of the line
<i>color</i> (optional)	Long specifying the color of the line Specify the default color using the of_SetDefaultColor function

Argument	Description
<i>backcolor</i> (optional)	Long specifying the background color of the line Specify the default background color using the of_SetDefaultBackColor function
<i>execute</i> (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 <i>execute</i> is TRUE, this function returns the result of the Modify statement that added the line; an empty string indicates success. If <i>execute</i> is FALSE, this function returns a string that you use as input to a Modify function to add the line.
Usage	<p>Depending on the <i>execute</i> argument, the function either adds the line to the DataWindow or returns syntax that you use in a Modify function to add the line.</p> <p>You can call this function to draw a separator line above the footer area when preparing a DataWindow for printing.</p> <p>Use Syntax 2 to add a line by specifying a VTextAlign enumerated data type.</p>
Examples	This example adds a line to the top of the DataWindow's footer:
	<pre>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</pre>

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

```
dwcontrolinstancename.of_AddLine ( band, penstyle, penwidth, placement
    , color, backcolor {, execute } )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)

Argument	Description
<i>band</i>	<p>String specifying the band to which the function will add the line.</p> <p>Valid bands are:</p> <ul style="list-style-type: none"> Header Detail Footer Summary Header.<i>levelnumber</i> Trailer.<i>levelnumber</i>
<i>penstyle</i>	<p>Linestyle enumerated data type specifying the pen style of the line:</p> <ul style="list-style-type: none"> Continuous! Dash! Dot! DashDot! DashDotDot! Transparent!
<i>penwidth</i>	Integer specifying the pen width of the line
<i>placement</i>	<p>VTextAlign enumerated data type specifying the line placement:</p> <ul style="list-style-type: none"> Bottom! MultiLine! Top! VCenter!
<i>color</i> (optional)	<p>Long specifying the color of the line</p> <p>Specify the default color using the <code>of_SetDefaultColor</code> function</p>
<i>backcolor</i> (optional)	<p>Long specifying the background color of the line</p> <p>Specify the default background color using the <code>of_SetDefaultBackColor</code> function</p>
<i>execute</i> (optional)	<p>Boolean specifying whether to execute the <code>Modify</code> function that adds the line to the band (TRUE) or return a string that you use to execute the <code>Modify</code> function (FALSE). The default is TRUE</p>

Return value	String. If <i>execute</i> is TRUE, this function returns the result of the <code>Modify</code> statement that added the line; an empty string indicates success. If <i>execute</i> is FALSE, this function returns a string that you use as input to a <code>Modify</code> function to add the line.
Usage	Depending on the <i>execute</i> argument, the function either adds the line to the DataWindow or returns syntax that you use in a <code>Modify</code> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>filename</i>	<p>String specifying the name of the bitmap file</p> <p>If the directory containing the file is not in your system path, you must specify the complete path</p>
<i>band</i>	<p>String specifying the name of the band to which the function will add the bitmap. Valid bands are:</p> <ul style="list-style-type: none"> • Header • Detail • Footer • Summary • Header.<i>levelnumber</i> • Trailer.<i>levelnumber</i>

Argument	Description
<i>xcoordinate</i>	<p>Either of the following:</p> <ul style="list-style-type: none"> • Integer specifying the <i>x</i> location of the bitmap. If you specify an integer for <i>xcoordinate</i>, you must also specify an integer for <i>ycoordinate</i> • Alignment enumerated data type (Center!, Justify!, Left!, or Right!). If you specify an alignment enumerated data type for <i>xcoordinate</i>, you must specify a VTextAlign enumerated data type for <i>ycoordinate</i>
<i>ycoordinate</i>	<p>Either of the following:</p> <ul style="list-style-type: none"> • Integer specifying the <i>y</i> location of the bitmap. If you specify an integer for <i>ycoordinate</i>, you must also specify an integer for <i>xcoordinate</i> • VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!). If you specify a VTextAlign enumerated data type for <i>ycoordinate</i>, you must specify an Alignment enumerated data type for <i>xcoordinate</i>
<i>border</i> (optional)	<p>Border enumerated data type specifying the border style:</p> <p>NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised!</p> <p>Specify the default border style using the of_SetDefaultBorder function</p>
<i>execute</i> (optional)	<p>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</p>

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

```
dwcontrolinstancename.of_AddText ( text, band, xcoordinate, ycoordinate
                                    {, border {, bold, italic, underline {, fontface, fontsize, characterset
                                    {, color, backcolor {, execute } } } } )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>text</i>	String specifying the text to be added
<i>band</i>	String specifying the band to which the function will add the text. Valid bands are: Header Detail Footer Summary Header.levelnumber Trailer.levelnumber
<i>xcoordinate</i>	Either of the following: <ul style="list-style-type: none"> • Integer specifying the <i>x</i> location of the text. If you specify an integer for <i>xcoordinate</i>, you must also specify an integer for <i>ycoordinate</i> • Alignment enumerated data type (Center!, Justify!, Left!, or Right!). If you specify an alignment enumerated data type for <i>xcoordinate</i>, you must specify a VTextAlign enumerated data type for <i>ycoordinate</i>

Argument	Description
<i>ycoordinate</i>	<p>Either of the following:</p> <ul style="list-style-type: none"> • Integer specifying the <i>y</i> location of the text. If you specify an integer for <i>ycoordinate</i>, you must also specify an integer for <i>xcoordinate</i> • VTextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!). If you specify a VTextAlign enumerated data type for <i>ycoordinate</i>, you must specify an Alignment enumerated data type for <i>xcoordinate</i>
<i>border</i> (optional)	<p>Border enumerated data type specifying the border style:</p> <p>NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised!</p> <p>Specify the default border style using the of_SetDefaultBorder function</p>
<i>bold</i> (optional)	<p>Boolean specifying whether text is bold:</p> <p>TRUE—Bold FALSE—Not bold (default)</p>
<i>italic</i> (optional)	<p>Boolean specifying whether text is italic:</p> <p>TRUE—Italic FALSE—Not italic (default)</p>
<i>underline</i> (optional)	<p>Boolean specifying whether text is underlined:</p> <p>TRUE—Underlined FALSE—Not underlined (default)</p>
<i>fontface</i> (optional)	<p>String specifying the font to use</p> <p>Specify the default font face using the of_SetDefaultFontFace function</p>
<i>fontsize</i> (optional)	<p>Integer specifying the point size of the font for the text</p> <p>Specify the default font size using the of_SetDefaultFontSize function</p>

Argument	Description
<i>charset</i> (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
<i>color</i> (optional)	Long specifying the color of the text Specify the default color using the of_SetDefaultColor function
<i>backcolor</i> (optional)	Long specifying the background color of the text Specify the default background color using the of_SetDefaultBackColor function
<i>execute</i> (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 <i>execute</i> is TRUE, this function returns the result of the Modify statement that added the text; an empty string indicates success. If <i>execute</i> is FALSE, this function returns a string that you use as input to a Modify function to add the text.
Usage	Depending on the <i>execute</i> argument, the function either adds the text to the DataWindow or returns syntax that you use in a Modify function to add the text.
Examples	This example adds text to the DataWindow's header area:
	<pre>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</pre>

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)
<i>type</i>	String specifying the type of object being added (text or computed column)
<i>text</i>	String specifying the text to be added
<i>band</i>	String specifying the band to which the function will add the text. Valid bands are: Header Detail Footer Summary Header.levelnumber Trailer.levelnumber
<i>xcoordinate</i>	Integer specifying the <i>x</i> location of the text
<i>ycoordinate</i>	Integer specifying the <i>y</i> location of the text
<i>height</i>	Integer specifying the height of the text
<i>width</i>	Integer specifying the width of the text
<i>border</i>	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
<i>bold</i>	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default)

Argument	Description
<i>italic</i>	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default)
<i>underline</i>	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default)
<i>fontface</i>	String specifying the font to use Specify the default font face using the of_SetDefaultFontFace function
<i>fontsize</i>	Integer specifying the point size of the font Specify the default font size using the of_SetDefaultFontSize function
<i>charerset</i>	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
<i>color</i>	Long specifying the color Specify the default color using the of_SetDefaultColor function
<i>backcolor</i>	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 *dwcontrolinstancename.of_CreateComposite (datawindows,
arrangevertical, trailfooter, slide, border)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)
<i>datawindows</i>	String array naming the DataWindow objects to make up the composite DataWindow
<i>arrangevertical</i>	Boolean specifying whether to arrange the DataWindows vertically (TRUE) or horizontally (FALSE)
<i>trailfooter</i>	String array of values to set the trail_footer property for each DataWindow. Valid values are: Yes No An empty string (defaults to Yes)
<i>slide</i>	String array of values to set the Slide properties for each DataWindow. If <i>arrangevertical</i> is TRUE, this function sets the DataWindow's SlideUp property. Valid values are: AllAbove DirectlyAbove No An empty string (defaults to AllAbove) If <i>arrangevertical</i> is FALSE, this function sets the SlideLeft property. Valid values are: Yes No
<i>border</i>	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 *dwcontrolinstancename.of_GetDefaultBackColor ()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

```
dwcontrolinstancename.of_GetDefaultBorder( )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrolinstancename.of_GetDefaultCharset()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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:
	<pre>FontCharSet lfc_charset lfc_charset = & dw_emp.inv_report.of_GetDefaultCharset() IF lfc_charset = ANSI! THEN ... </pre>

of_GetDefaultColor

Description	Retrieves the default color.
Access	Public
Syntax	<code>dwcontrolinstancename.of_GetDefaultColor()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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:
	<pre>Long ll_color ll_color = & dw_emp.inv_report.of_GetDefaultColor()</pre>

of_GetDefaultFontFace

Description Retrieves the default font (for example, Arial).

Access Public

Syntax dwcontrol.*instancename*.**of_GetDefaultFontFace** ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrol.instancename.of_GetInfo (infoobject)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default for this is inv_report)
<i>infoobject</i>	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:
	<pre>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)</pre>

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	<code>dwcontrol.instancename.of_GetPictureSize (filename, height, width)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>filename</i>	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
<i>height</i>	Integer specifying the height of the picture in PBUs or pixels (passed by reference)
<i>width</i>	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	<i>dwcontrol.instancename.of_GetPropertyInfo (propertyobject)</i>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default for this is <i>inv_report</i>)
<i>propertyobject</i>	<i>n_cst_propertyattrib</i> 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 ofGetPropertyInfo function:
----------	--

```
n_cst_propertyattrib  lnv_prop

dw_1.inv_report.ofGetPropertyInfo(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	<code>dwcontrol.instancename.of_GetTextSizePos (text, band, halign, valign, bold, italic, underline, fontface, fontsize, height, width, xcoordinate, ycoordinate)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>text</i>	String specifying the text to be added
<i>band</i>	String specifying the band. Valid values are: Header Detail Footer Summary Header. <i>levelnumber</i> Trailer. <i>levelnumber</i>
<i>halign</i>	Alignment enumerated data type (Center!, Justify!, Left!, or Right!) specifying horizontal alignment
<i>valign</i>	V TextAlign enumerated data type (Bottom!, MultiLine!, Top!, or VCenter!) specifying vertical alignment
<i>bold</i>	Boolean specifying whether text is bold: TRUE—Bold FALSE—Not bold (default)
<i>italic</i>	Boolean specifying whether text is italic: TRUE—Italic FALSE—Not italic (default)
<i>underline</i>	Boolean specifying whether text is underlined: TRUE—Underlined FALSE—Not underlined (default)
<i>fontface</i>	String specifying the font to use. Specify the default font face using the of_SetDefaultFontFace function
<i>fontsize</i>	Integer specifying the point size of the font. Specify the default font size using the of_SetDefaultFontSize function
<i>height</i>	Integer specifying the height of the text
<i>width</i>	Integer specifying the width of the text
<i>xcoordinate</i>	Integer specifying the <i>x</i> location of the text
<i>ycoordinate</i>	Integer specifying the <i>y</i> location of the text

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

Usage Internal.

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 *dwcontrolinstancename.of_GetUndoLevels()*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)

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 *dwcontrolinstancename.of_PrepPrint(underlineheadings)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control

Argument	Description
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>underlineheadings</i>	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	<p>Call this function to modify a DataWindow to a format more suitable for printing (black on white; no borders around columns).</p> <p>This function removes borders from all columns, including headers. You can highlight headings by setting <i>underlineheadings</i> to TRUE, which replaces header borders with underlines.</p> <p>This function is called by of_PrintReport and calls of_SetColor and of_SetBorder.</p>
Examples	<p>This example calls of_PrepPrint before printing dw_employee using underlined headings:</p> <pre> 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") ENDIF // Print report. dw_employee.Print(TRUE) </pre>
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	<code>dwcontrolinstancename.of_PrintReport (<i>underlineheadings</i>, {, <i>canceledialog</i> })</code>										
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)</td></tr><tr><td><i>underlineheadings</i></td><td>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</td></tr><tr><td><i>canceledialog</i> (optional)</td><td>Boolean specifying whether to display the Cancel dialog box (TRUE) or not (FALSE). The default is FALSE</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)	<i>underlineheadings</i>	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	<i>canceledialog</i> (optional)	Boolean specifying whether to display the Cancel dialog box (TRUE) or not (FALSE). The default is FALSE
Argument	Description										
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control										
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)										
<i>underlineheadings</i>	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										
<i>canceledialog</i> (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: <pre>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</pre>										

of_ResetUndo

Description	Resets the number of undo levels to zero.						
Access	Public						
Syntax	<code>dwcontrolinstancename.of_ResetUndo ()</code>						
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)</td></tr></tbody></table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
Argument	Description						
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control						
<i>instancename</i>	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

```
dwcontrolinstancename.of_SetBackground ( filename {, xcoordinate  
{, ycoordinate {, height, width {, execute } } } })
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>filename</i>	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
<i>xcoordinate</i> (optional)	Integer specifying the <i>x</i> location of the bitmap. The default is 0
<i>ycoordinate</i> (optional)	Integer specifying the <i>y</i> location of the bitmap. The default is 0
<i>height</i> (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
<i>width</i> (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
<i>execute</i> (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 <i>execute</i> is TRUE, this function returns the result of the Modify statement that added the background picture; an empty string indicates success. If <i>execute</i> 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.
Examples	Depending on the <i>execute</i> 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. This example uses the <i>of_SetBackground</i> 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

Description

Add a background picture, sizing it to fit the DataWindow

Add a background picture, optionally sizing it to fit the DataWindow.

Access

Public

Syntax

```
dwcontrolinstancename.of_SetBackground ( filename , sizetofit,
                                         xcoordinate, ycoordinate, height, width execute )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>filename</i>	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
<i>sizetofit</i>	Boolean specifying whether to set the picture width to the width of the DataWindow and use the <i>height/width</i> ratio to set the height (TRUE) or use the <i>height</i> and <i>width</i> values
<i>xcoordinate</i>	Integer specifying the <i>x</i> location of the bitmap
<i>ycoordinate</i>	Integer specifying the <i>y</i> location of the bitmap

Argument	Description
<i>height</i>	Integer specifying the height of the bitmap being placed. If you pass 0, the function uses the of_GetPictureSize function to calculate this value
<i>width</i>	Integer specifying the width of the bitmap being placed. If you pass 0, the function uses the of_GetPictureSize function to calculate this value
<i>execute</i>	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 ***dwcontrolinstancename.of_SetBorder (border {, objectname {, band {, execute } } })***

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)
<i>border</i>	Border enumerated data type specifying the border style: NoBorder! ShadowBox! Box! ResizeBorder! Underline! Lowered! Raised! Specify the default border style using the <i>of_SetDefaultBorder</i> function
<i>objectname</i> (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 (*)
<i>band</i> (optional)	String specifying the band in which to change the objects. The function ignores this argument if you pass a name in <i>objectname</i> . Valid values are: An asterisk (*), which changes all objects in the DataWindow Header Detail Footer Summary Header. <i>levelnumber</i> Trailer. <i>levelnumber</i> The default is an asterisk (*)
<i>execute</i> (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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>color</i>	Long specifying a long value that identifies the new color. If you pass a NULL, the function does not change the color
<i>backgroundcolor</i>	Long specifying a long value that identifies the new background color. If you pass a NULL, the function does not change the background color
<i>objectname</i> (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
<i>band</i> (optional)	<p>String specifying the band in which to change the objects. The function ignores this argument if you pass a name in <i>objectname</i>. Valid bands are:</p> <p>An asterisk (*), which changes colors for all objects Header Detail Footer Summary Header.<i>levelnumber</i> Trailer.<i>levelnumber</i></p> <p>The default is an asterisk (*)</p>
<i>execute</i> (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

`dwcontrolinstancename.of_SetDefaultBackColor (color)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>color</i>	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

`dwcontrolinstancename.of_SetDefaultBorder (border)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>border</i>	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	<code>dwcontrol.instancename.of_SetDefaultCharset (charset)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>charset</i>	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	<code>dwcontrol.instancename.of_SetDefaultColor (color)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>color</i>	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 dwcontrolinstancename.of_SetDefaultFontFace (*font*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>font</i>	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 dwcontrolinstancename.of_SetDefaultFontSize (*fontsize*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>fontsize</i>	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

Description

Set the font, including bold, italic, and underline

Sets the font, including bold, italic, and underline.

Access

Public

Syntax

```
dwcontrolinstancename.of_SetFont ( fontface, fontsize, fontfamily, fontpitch  
{, bold, italic, underline {, objectname {, band {, execute } } } } )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>fontface</i>	String specifying the new font. If you pass a NULL, the function does not change the font
<i>fontsize</i>	Integer specifying the point size of the new font. If you pass a NULL, the function does not change the font size
<i>fontfamily</i>	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
<i>fontpitch</i>	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
<i>bold</i> (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

Argument	Description
<i>italic</i> (optional)	<p>Boolean specifying whether text is italic:</p> <ul style="list-style-type: none"> TRUE—Italic FALSE—Not italic (default) <p>If you pass a NULL, the function does not change the italic property</p>
<i>underline</i> (optional)	<p>Boolean specifying whether text is underlined:</p> <ul style="list-style-type: none"> TRUE—Underlined FALSE—Not underlined (default) <p>If you pass a NULL, the function does not change the underline property</p>
<i>objectname</i> (optional)	<p>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</p> <p>The default is an asterisk (*)</p>
<i>band</i> (optional)	<p>String specifying the band in which to change the objects. The function ignores this argument if you pass a name in <i>objectname</i>. Valid bands are:</p> <ul style="list-style-type: none"> An asterisk (*), which changes the border for all objects Header Detail Footer Summary Header.levelnumber Trailer.levelnumber <p>The default is an asterisk (*)</p>
<i>execute</i> (optional)	<p>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</p>

Return value	String. If <i>execute</i> is TRUE, this function returns the result of the Modify statement that changes the font; an empty string indicates success. If <i>execute</i> is FALSE, this function returns a string that you use as input to a Modify function to change the font.
Usage	<p>Call this function to change the font for a DataWindow column before printing.</p> <p>Depending on the <i>execute</i> argument, the function either changes the font or returns syntax that you use in a Modify function to change the font.</p> <p>Use Syntax 2 to call this function, leaving bold, italic, and underline properties as is.</p>

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)
<i>fontface</i>	String specifying the new font. If you pass a NULL, the function does not change the font
<i>fontsize</i>	Integer specifying the point size of the new font. If you pass a NULL, the function does not change the font size
<i>fontfamily</i>	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

Argument	Description
<i>fontpitch</i>	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
<i>objectname</i> (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 (*)
<i>band</i> (optional)	String specifying the band in which to change the objects. The function ignores this argument if you pass a name in <i>objectname</i> . 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)
<i>zoompercentage</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_report</i> (the u_dw default is <i>inv_report</i>)

Argument	Description
<i>band</i>	String specifying the band to shift. Valid bands are: An asterisk (*), which shifts entire the DataWindow Header Detail Footer Summary Header. <i>levelnumber</i> Trailer. <i>levelnumber</i>
<i>xunits</i>	Integer specifying the number of units to shift the objects right (to shift left, pass a negative number)
<i>yunits</i>	Integer specifying the number of units to shift the objects down (to shift up, pass a negative number)
<i>execute</i> (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	<code>dwcontrolinstancename.of_Undo (levelstoundo)</code>								
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><code>dwcontrol</code></td><td>Instance name of the u_dw-based DataWindow control</td></tr><tr><td><code>instancename</code></td><td>Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)</td></tr><tr><td><code>levelstoundo</code></td><td>Integer specifying the number of changes to restore. Specify 0 to undo all changes</td></tr></tbody></table>	Argument	Description	<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control	<code>instancename</code>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)	<code>levelstoundo</code>	Integer specifying the number of changes to restore. Specify 0 to undo all changes
Argument	Description								
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control								
<code>instancename</code>	Instance name of n_cst_dwsrv_report (the u_dw default is inv_report)								
<code>levelstoundo</code>	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	<p>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.</p> <p>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 <i>execute</i> set to FALSE, for input to a single Modify function. To remove these changes, call of_Undo(3).</p> <p>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.</p>								
Examples	<p>This example uses the of_Undo function after printing a DataWindow:</p> <pre>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</pre>								

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	<pre> graph TD pfc_n_base[pfc_n_base] --> n_base[n_base] n_base --> ObjectFuncs1[Object functions] ObjectFuncs1 --> Protected[Protected of_MessageBox] Protected --> pfc_n_cst_dwsrv[pfc_n_cst_dwsrv] pfc_n_cst_dwsrv --> n_cst_dwsrv[n_cst_dwsrv] n_cst_dwsrv --> ObjectFuncs2[Object functions] </pre> <p>The diagram illustrates the object hierarchy for the <code>n_cst_dwsrv_reqcolumn</code> class. It starts with <code>pfc_n_base</code>, which contains <code>n_base</code>. <code>n_base</code> contains <code>Object functions</code>, which in turn contain <code>Protected of_MessageBox</code>. This leads to <code>pfc_n_cst_dwsrv</code>, which contains <code>n_cst_dwsrv</code>. Finally, <code>n_cst_dwsrv</code> contains another set of <code>Object functions</code>.</p>

Library	PFCDWSRV.PBL PFEDWSRV.PBL
Object relationships	<code>n_cst_infoattrib</code> <code>n_cst_propertyattrib</code>
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
ofGetPropertyInfo	of_SetNotNullIfEmpty
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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_reqcolumn</i> (the u_dw default is <i>inv_reqcolumn</i>)

Argument	Description
<i>infoobject</i>	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:
	<pre>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)</pre>

of_GetPropertyInfo

Description	Retrieves information about the service's properties.
Access	Public
Syntax	<i>dwcontrolinstancename.ofGetPropertyInfo (propertyobject)</i>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
<i>propertyobject</i>	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 ofGetPropertyInfo function:

```
n_cst_propertyattrib lnv_prop

dw_1.inv_reqcolumn.ofGetPropertyInfo(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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
<i>columns</i>	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

dwcontrolinstancename.of_IsRegistered (column)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
<i>column</i>	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	<code>dwcontrolinstancename.of_RegisterSkipColumn (column)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_reqcolumn</i> (the u_dw default is <i>inv_reqcolumn</i>)
<i>column</i>	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_SetNullIfEmpty

Description Replaces empty strings with a null value, suppressing the DataWindow's Value Required dialog box.

Access Public

Syntax `dwcontrolinstancename.of_SetNullIfEmpty (row, column, text)`

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_reqcolumn</i> (the u_dw default is <i>inv_reqcolumn</i>)
<i>row</i>	Long specifying the row to check
<i>column</i>	String specifying the column to check
<i>text</i>	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

dwcontrolinstancename.of_UnRegister (column)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
<i>column</i>	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

n_cst_dwsrv_reqcolumn

Syntax	<code>dwcontrol.<i>instancename</i>.of_UpdateSkipString (<i>skipcolumns</i>)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_reqcolumn (the u_dw default is inv_reqcolumn)
<i>skipcolumns</i>	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.

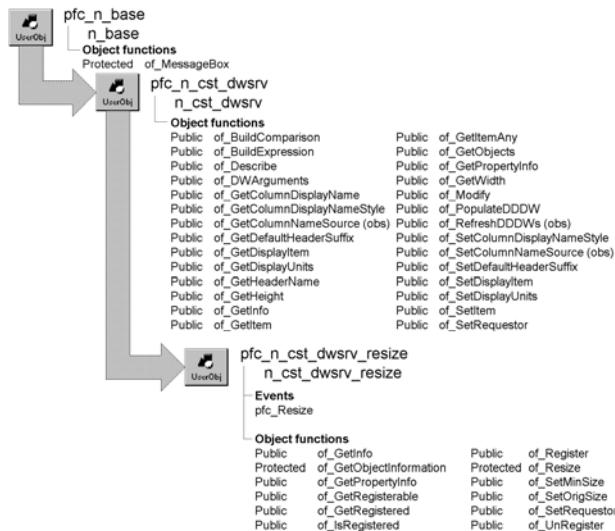
Examples This example is from the of_RegisterSkipColumn function:

```
...
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.

This service provides two resizing options:

- **For simple resizing** Call the of_Register function passing n_cst_dwsrv_resize constants, such as FIXEDBOTTOM
- **For total control over resizing** Implement weighted resize by calling the of_Register function with explicit specifications for moving and scaling

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.

PFC 7 custom class extension

To further the extendibility of PFC, a new n_cst_dwsrv_resizeattrib custom class user object replaces the os_resize object structure that is now obsolete.

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 FixedToLeft	String	Public	Use with of_Register
FIXEDRIGHT_SCALEBOTTOM	Constant set to FixedToLeft&ScaleToBottom	String	Public	Use with of_Register
FIXEDRIGHTBOTTOM	Constant set to FixedToLeft&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
inv_registered []	Array of registered columns and text objects	n_cst_dws rv_resizeat trib	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 *dwcontrolinstancename.EVENT pfc_Resize (sizetype, newwidth, newheight)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>sizetype</i>	UnsignedLong specifying the resize type: <ul style="list-style-type: none">• 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
<i>newwidth</i>	Integer specifying the new DataWindow width
<i>newheight</i>	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
ofGetPropertyInfo	of_SetMinSize
ofGetRegisterable	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>infoobject</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>control</i>	String specifying the object for which the function returns information

Argument	Description
<i>type</i>	String into which the function places the object type Object types include column text, bitmap, compute, and so on (passed by reference)
<i>x</i>	Integer into which the function places the x coordinate (passed by reference)
<i>y</i>	Integer into which the function places the y coordinate (passed by reference)
<i>width</i>	Integer into which the function places the object's width (passed by reference)
<i>height</i>	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)
...

```

ofGetPropertyInfo

Description Retrieves information about the service's properties.

Access Public

Syntax *dwcontrolinstancename.ofGetPropertyInfo (propertyobject)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>propertyobject</i>	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.ofGetPropertyInfo(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

```
dwcontrolinstancename.of_GetRegisterable ( columns )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>columns</i>	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.

n_cst_dwsrv_resize

Access	Public
Syntax	<code>dwcontrol.instancename.of_GetRegistered (object, scale, movex, movey, scalewidth, scaleheight)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>object</i>	DWObject or string specifying the object for which information is returned
<i>scale</i>	Boolean into which the function places the object's scale switch (passed by reference)
<i>movex</i>	Integer into which the function places the object's move x specification (passed by reference)
<i>movey</i>	Integer into which the function places the object's move y specification (passed by reference)
<i>scalewidth</i>	Integer into which the function places the object's scale width specification (passed by reference)
<i>scaleheight</i>	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	<code>dwcontrol.instancename.of_IsRegistered (column)</code>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>column</i>	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

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
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of <code>n_cst_dwsrv_resize</code> (the u_dw default is <code>inv_resize</code>)
<code>object</code>	DWObject variable specifying the object to be resized
<code>movex</code>	Integer specifying the percentage to move the object along the x axis

Argument	Description
<i>movey</i>	Integer specifying the percentage to move the object along the y axis
<i>scalewidth</i>	Integer specifying the percentage to resize the object along the x axis
<i>scaleheight</i>	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.of_Retrieve()
```

Syntax 2

Description

Specify a DWObject variable and resize method

Registers an object for resizing, given a DWObject variable and a resize method.

Access

Public

Syntax

dwcontrolinstancename.of_Register (object, resizemethod)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_resize</i> (the u_dw default is <i>inv_resize</i>)
<i>object</i>	DWObject variable specifying the object to be resized

Argument	Description
<i>resizemethod</i>	<p>String or n_cst_dwsrv_resize constant specifying the action the service takes when the DataWindow Resize event is triggered:</p> <ul style="list-style-type: none"> • "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.of_Retrieve()
```

Syntax 3

Description

Specify a string and resize percentages

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>object</i>	String specifying the object to be resized

Argument	Description
<i>movex</i>	Integer specifying the percentage to move the object along the x axis
<i>movey</i>	Integer specifying the percentage to move the object along the y axis
<i>scalewidth</i>	Integer specifying the percentage to resize the object along the x axis
<i>scaleheight</i>	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 of_Retrieve()
```

Syntax 4

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_resize</i> (the u_dw default is <i>inv_resize</i>)
<i>object</i>	String specifying the object to be resized

Argument	Description
<i>resizemethod</i>	String or n_cst_dwsrv_resize constant specifying the action the service takes when the DataWindow Resize event is triggered: <ul style="list-style-type: none"> • "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: <pre>this.inv_resize.of_Register & ("comments", inv_resize.SCALEBOTTOM) this.of_SetTransObject(SQLCA) this.of_Retrieve()</pre>

of_Resize

Description	Moves or resizes registered objects as specified in the of_Register function.
Access	Protected
Syntax	<i>dwcontrol.instancename.of_Resize (newwidth, newheight)</i>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>newwidth</i>	Integer specifying the new DataWindow width
<i>newheight</i>	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

```
dwcontrolinstancename.of_SetMinSize ( minimumwidth, minimumheight )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_resize</i> (the u_dw default is <i>inv_resize</i>)
<i>minimumwidth</i>	Integer specifying minimum width
<i>minimumheight</i>	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

```
dwcontrolinstancename.of_SetOrigSize ( width, height )
```

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_resize</i> (the u_dw default is <i>inv_resize</i>)
<i>width</i>	Integer specifying the DataWindow width
<i>height</i>	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

dwcontrolinstancename.of_SetRequestor (requestor)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>requestor</i>	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

Description

Specify a DWObject

Removes an object from the list of registered objects, given a DWObject variable.

Access

Public

Syntax

dwcontrolinstancename.of_UnRegister (object)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>object</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_resize (the u_dw default is inv_resize)
<i>object</i>	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" )
```

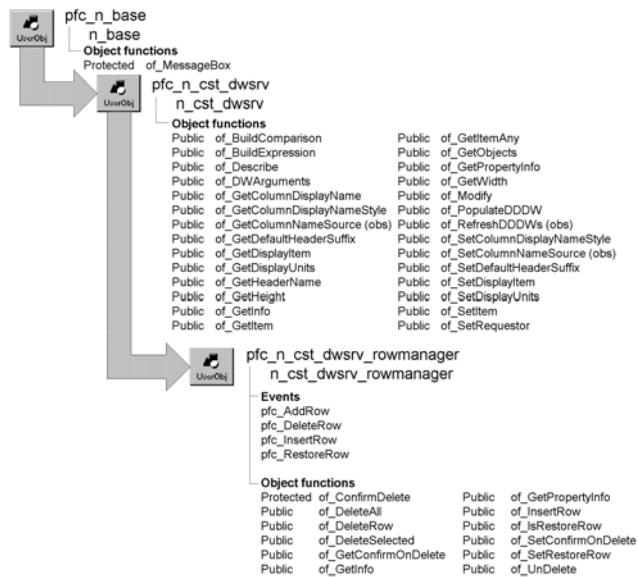
n_cst_dwsrv_rowmanager

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager).
<i>nextrow</i>	Long specifying the row before which the event adds a row. From within the pfc_InsertRow event, access this value through the <i>al_beforerow</i> 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:

<code>of_ConfirmDelete</code>	<code>of_GetPropertyInfo</code>
<code>of_DeleteAll</code>	<code>of_InsertRow</code>
<code>of_DeleteRow</code>	<code>of_IsRestoreRow</code>
<code>of_DeleteSelected</code>	<code>of_SetConfirmOnDelete</code>
<code>of_GetConfirmOnDelete</code>	<code>of_SetRestoreRow</code>
<code>of_GetInfo</code>	<code>of_UnDelete</code>

of_ConfirmDelete

Description	Displays a message box, prompting the user to confirm a deletion request.								
Access	Protected								
Syntax	<code>dwcontrolinstancename.of_ConfirmDelete (amount)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)</td></tr> <tr> <td><i>amount</i></td><td>Long specifying the number of rows to be deleted</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)	<i>amount</i>	Long specifying the number of rows to be deleted
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)								
<i>amount</i>	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	<code>dwcontrolinstancename.of_DeleteAll ()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
Argument	Description						
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control						
<i>instancename</i>	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

dwcontrolinstancename.of_DeleteRow (rownumber)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_rowmanager</i> (the u_dw default is <i>inv_rowmanager</i>)
<i>rownumber</i>	Long specifying the row to be deleted. If <i>rownumber</i> 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	<code>dwcontrolinstancename.of_DeleteSelected()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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:
	<pre>Integer li_return li_return = & dw_emplist.inv_rowmanager.of_DeleteSelected() ...</pre>

of_GetConfirmonDelete

Description	Reports whether the service prompts the user to confirm deletion requests.
Access	Public
Syntax	<code>dwcontrolinstancename.of_GetConfirmonDelete()</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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:
	<pre>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))</pre>

of_GetInfo

Description	Retrieves service information.								
Access	Public								
Syntax	<code>dwcontrolinstancename.of_GetInfo (infoobject)</code>								
	<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th><th style="text-align: center;">Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)</td></tr> <tr> <td><i>infoobject</i></td><td>N_cst_infoattrib instance into which the function places service information (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)	<i>infoobject</i>	N_cst_infoattrib instance into which the function places service information (passed by reference)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)								
<i>infoobject</i>	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)

```

ofGetPropertyInfo

Description	Retrieves information about the service's properties.								
Access	Public								
Syntax	<code>dwcontrolinstancename.ofGetPropertyInfo (propertyobject)</code>								
	<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th><th style="text-align: center;">Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)</td></tr> <tr> <td><i>propertyobject</i></td><td>N_cst_propertyattrib instance into which the function places property information (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)	<i>propertyobject</i>	N_cst_propertyattrib instance into which the function places property information (passed by reference)
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)								
<i>propertyobject</i>	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.ofGetPropertyInfo(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

dwcontrolinstancename.of_InsertRow (insertionpoint)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowmanager (the u_dw default is inv_rowmanager)
<i>insertionpoint</i>	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

dwcontrolinstancename.of_IsRestoreRow ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_rowmanager</i> (the u_dw default is <i>inv_rowmanager</i>)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_rowmanager</i> (the u_dw default is <i>inv_rowmanager</i>)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
    nv_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:

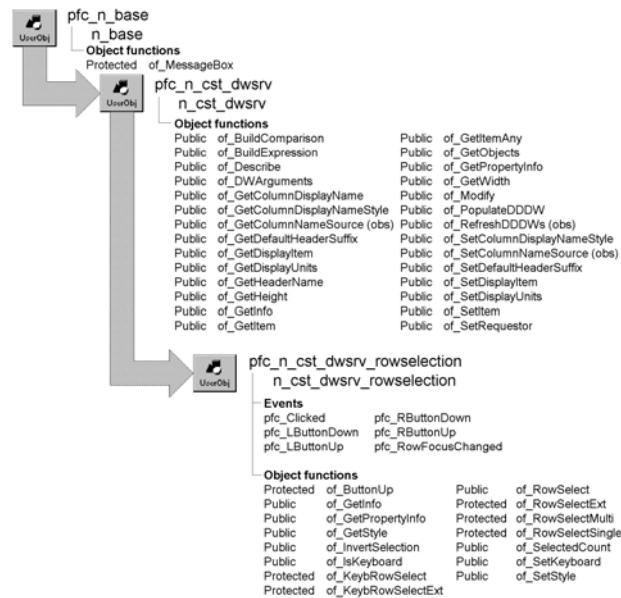
- **Single-row selection** Handles row selection when your DataWindow allows one row to be selected at a time.
- **Multirow selection** Handles row selection by allowing your runtime users to select multiple rows with single clicks. These rows can be contiguous or noncontiguous.

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

Selecting rows in extended mode

If you choose the extended mode for a tabular list style DataWindow, the mouse pointer must be somewhere inside the DataWindow in order to select rows with the space bar.

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
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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_Clicked event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_Clicked event, access this value through the <i>ai_ypos</i> argument
<i>row</i>	Long indicating the clicked row. From within the pfc_Clicked event, access this value through the <i>al_row</i> argument
<i>dwobject</i>	DWOObject variable indicating the clicked object (passed by reference). From within the pfc_Clicked event, access this value through the <i>adwo_obj</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)

Argument	Description
<i>flags</i>	<p>Unsigned long indicating the modifier keys and mouse buttons that are pressed. Values are:</p> <ul style="list-style-type: none"> 1—Left mouse button 2—Right mouse button 4—SHIFT key 8—CTRL key 16—Middle mouse button <p>From within the pfc_LButtonDown event, access this value through the <i>aul_flags</i> argument</p>
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_LButtonDown event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_LButtonDown event, access this value through the <i>ai_ypos</i> 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

dwcontrolinstancename.Event **pfc_LButtonUp** (*flags*, *x*, *y*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>flags</i>	<p>Unsigned Long indicating the modifier keys and mouse buttons that are pressed. Values are:</p> <ul style="list-style-type: none"> 1—Left mouse button 2—Right mouse button 4—SHIFT key 8—CTRL key 16—Middle mouse button <p>From within the pfc_LButtonUp event, access this value through the <i>aul_flags</i> argument</p>
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_LButtonUp event, access this value through the <i>ai_xpos</i> argument

Argument	Description
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_LButtonUp event, access this value through the <i>ai_ypos</i> 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 *dwcontrolinstancename*.Event **pfc_RButtonDown** (*flags*, *x*, *y*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>flags</i>	<p>Unsigned Long indicating the modifier keys and mouse buttons that are pressed. Values are:</p> <ul style="list-style-type: none"> 1—Left mouse button 2—Right mouse button 4—SHIFT key 8—CTRL key 16—Middle mouse button <p>From within the pfc_RButtonDown event, access this value through the <i>aul_flags</i> argument</p>
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_RButtonDown event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_RButtonDown event, access this value through the <i>ai_ypos</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>flags</i>	<p>Unsigned Long indicating the modifier keys and mouse buttons that are pressed. Values are:</p> <ul style="list-style-type: none"> 1—Left mouse button 2—Right mouse button 4—SHIFT key 8—CTRL key 16—Middle mouse button <p>From within the pfc_RButtonUp event, access this value through the <i>aul_flags</i> argument</p>
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_RButtonUp event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_RButtonUp event, access this value through the <i>ai_ypos</i> 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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>row</i>	Long specifying the new row. From within the pfc_RowFocusChanged event, access this value through the <i>al_row</i> 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:

<code>of_ButtonUp</code>	<code>of_RowSelect</code>
<code>of_GetInfo</code>	<code>of_RowSelectExt</code>
<code>ofGetPropertyInfo</code>	<code>of_RowSelectMulti</code>
<code>of_getStyle</code>	<code>of_RowSelectSingle</code>
<code>of_InvertSelection</code>	<code>of_SelectedCount</code>
<code>of_IsKeyboard</code>	<code>of_SetKeyboard</code>
<code>of_KeybRowSelect</code>	<code>of_SetStyle</code>
<code>of_KeybRowSelectExt</code>	

of_ButtonUp

Description Performs extended selection processing.

Access Protected

Syntax `dwcontrolinstancename.of_ButtonUp ()`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of <code>n_cst_dwsrv_rowselection</code> (the u_dw default is <code>inv_rowselect</code>)

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 `dwcontrolinstancename.of_GetInfo (infoobject)`

Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of <code>n_cst_dwsrv_rowselection</code> (the u_dw default is <code>inv_rowselect</code>)

Argument	Description
<i>infoobject</i>	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)
```

ofGetPropertyInfo

Description	Retrieves information about the service's properties.
Access	Public
Syntax	<i>dwcontrolinstancename.ofGetPropertyInfo (propertyobject)</i>

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>propertyobject</i>	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 ofGetPropertyInfo function:

```
n_cst_propertyattrib lnv_prop

dw_1.inv_rowselect.ofGetPropertyInfo(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	<code>dwcontrolinstancename.of_GetStyle()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
Argument	Description						
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control						
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)						
Return value	<p>Integer. Returns one of the following values:</p> <ul style="list-style-type: none"> • 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:						
	<pre>IF IsValid(inv_rowselect) THEN IF inv_rowselect.of_GetStyle() = 2 THEN ... END IF END IF</pre>						

of_InvertSelection

Description	Reverses selection on all rows. Selected rows become unselected; unselected rows become selected.						
Access	Public						
Syntax	<code>dwcontrolinstancename.of_InvertSelection()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
Argument	Description						
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control						
<i>instancename</i>	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:						
	<pre>dw_emplist.inv_rowselect.of_InvertSelection()</pre>						

of_IsKeyboard

Description Reports whether keyboard support is enabled.

Access Public

Syntax **dwcontrol.instancename.of_IsKeyboard ()**

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)
<i>row</i>	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	<code>dwcontrol.instancename.of_KeybRowSelectExt (row, ctrlpressed, shiftpressed)</code>
Argument	Description
<code>dwcontrol</code>	Instance name of the u_dw-based DataWindow control
<code>instancename</code>	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)
<code>row</code>	Long specifying the current row number
<code>ctrlpressed</code>	Boolean indicating whether the CTRL key is down
<code>shiftpressed</code>	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: <ul style="list-style-type: none"> • Single-row selection of_RowSelectSingle • Multirow selection of_RowSelectMulti • Extended selection of_RowSelectExt
Access	Protected

Syntax	<code>dwcontrol.instancename.of_RowSelect (clickedrow)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>clickedrow</i>	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:
	<pre>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)</pre>

of_RowSelectExt

Description	Selects or clears rows depending on whether the CTRL and SHIFT keys are down.
Access	Protected
Syntax	<code>dwcontrol.instancename.of_RowSelectExt (clickedrow, ctrlpressed, shiftpressed)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>clickedrow</i>	Long containing the number of the clicked row
<i>ctrlpressed</i>	Boolean indicating whether the CTRL key is down
<i>shiftpressed</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>clickedrow</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>clickedrow</i>	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 dwcontrolinstancename.**of_SelectedCount** (*selectedrows*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>selectedrows</i>	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 dwcontrolinstancename.**of_SetKeyboard** (*boolean*)

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default for this value is inv_rowselect)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_rowselection (the u_dw default is inv_rowselect)
<i>selectoption</i>	Integer or n_cst_dwsrv_rowselection constant specifying the row selection option: <ul style="list-style-type: none"> • 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)
```

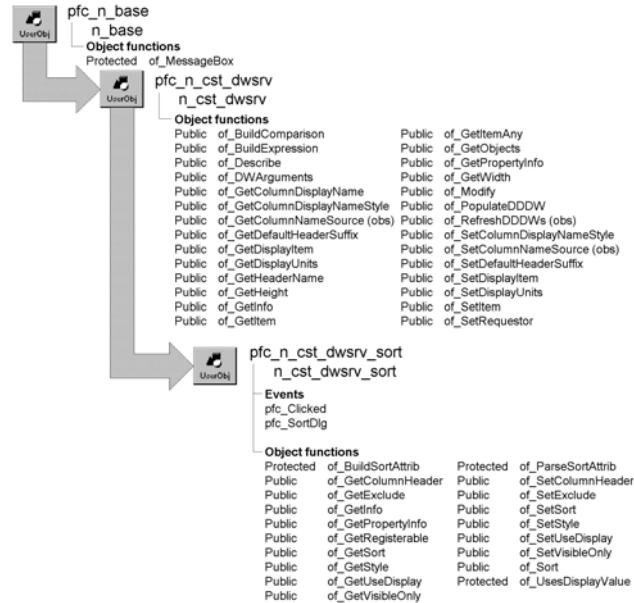
n_cst_dwsrv_sort

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)
```

Sorting by column headers

Columns that you add to a DataWindow are not automatically assigned column headers. If you are sorting columns by column header, make sure that all columns added to the DataWindow have headers, and that these conform to the naming scheme for headers. The default naming scheme uses the suffix _t, but you can change this by calling the of_SetDefaultHeaderSuffix function.

- 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
DROPOWNLISTBOX	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
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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>x</i>	Integer indicating the <i>x</i> position of the mouse click. From within the pfc_Clicked event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer indicating the <i>y</i> position of the mouse click. From within the pfc_Clicked event, access this value through the <i>ai_ypos</i> argument
<i>row</i>	Long indicating the clicked row. From within the pfc_Clicked event, access this value through the <i>al_row</i> argument
<i>dwobject</i>	DWOObject variable indicating the clicked object (passed by reference). From within the pfc_Clicked event, access this value through the <i>adwo_obj</i> 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_SortDig**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 pre-coded object functions:

of_BuildSortAttrib	of_ParseSortAttrib
of_GetColumnHeader	of_SetColumnHeader
of_GetExclude	of_SetExclude
of_GetInfo	of_SetSort
ofGetPropertyInfo	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)
<i>sortattrib</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)

Argument	Description
<i>infoobject</i>	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)
```

ofGetPropertyInfo

- Description Retrieves information about the service's properties.
- Access Public
- Syntax *dwcontrolinstancename.ofGetPropertyInfo (propertyobject)*

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>propertyobject</i>	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 ofGetPropertyInfo function:

```
n_cst_propertyattrib lnv_prop

dw_1.inv_sort.ofGetPropertyInfo(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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>columns</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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

dwcontrolinstancename.of_GetStyle ()

Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)

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
- **DROPODOWNLISTBOX 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
        ...
    END IF
```

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	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	<code>dwcontrol.instancename.of_ParseSortAttrib (originalsort, sortattrib)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)
<i>originalsort</i>	String containing the original sort statement
<i>sortattrib</i>	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 <i>n_cst_dwsrv_sort</i> sorts when the user clicks on a column header. With column header sorting (a feature found in many Windows-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	<code>dwcontrol.instancename.of_SetColumnHeader (boolean)</code>
Argument	Description
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)
<i>boolean</i>	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.
Usage	The N-Up presentation style does not support column header sorting.

If your DataWindow object uses column header suffixes other than _t, you must first call the n_cst_dwsrv of _SetDefaultHeaderSuffix function to specify the column header suffix.

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>excludecols</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>sort</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)
<i>sortstyle</i>	Integer or <i>n_cst_dwsrv_sort</i> constant specifying the sort style: <ul style="list-style-type: none">• DEFAULT or 0 PowerBuilder Sort dialog box (default)• DRAGDROP or 1 W_sortdragdrop dialog box• SIMPLE or 2 W_sortsingle dialog box• DROPODOWNLISTBOX 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-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):								
	<pre>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)</pre>								
of_SetUseDisplay									
Description	Specifies whether PFC sorts on display values or data values, for columns that display using a code table.								
<hr/> <p>PowerBuilder Sort dialog box always uses data values The PowerBuilder Sort dialog box always uses data values for sorting.</p> <hr/>									
Access	Public								
Syntax	<code>dwcontrolinstancename.of_SetUseDisplay (boolean)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>dwcontrol</i></td><td>Instance name of the u_dw-based DataWindow control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)</td></tr> <tr> <td><i>boolean</i></td><td>Boolean specifying whether to use display values (TRUE) or data values (FALSE) for sorting</td></tr> </tbody> </table>	Argument	Description	<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control	<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)	<i>boolean</i>	Boolean specifying whether to use display values (TRUE) or data values (FALSE) for sorting
Argument	Description								
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control								
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)								
<i>boolean</i>	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):								
	<pre>dw_emplist.of_SetSort(TRUE) dw_emplist.of_SetStyle(dw_emplist.inv_sort.SIMPLE) dw_emplist.inv_sort.of_SetUseDisplay(TRUE)</pre>								
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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)
<i>boolean</i>	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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of <i>n_cst_dwsrv_sort</i> (the u_dw default is <i>inv_sort</i>)

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
<i>dwcontrol</i>	Instance name of the u_dw-based DataWindow control
<i>instancename</i>	Instance name of n_cst_dwsrv_sort (the u_dw default is inv_sort)
<i>column</i>	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 + ") "
...

```

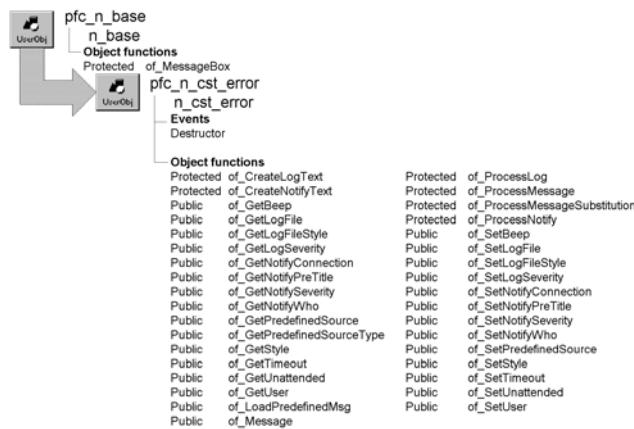
n_cst_error

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\veiserr.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
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

Instance variable	Description	Data type	Access	Usage
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
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

Instance variable	Description	Data type	Access	Usage
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	<i>application.instancename.of_CreateLogText (text)</i>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>application</i></td><td>Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)</td></tr> <tr> <td><i>text</i></td><td>String containing text to be logged</td></tr> </tbody> </table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)	<i>text</i>	String containing text to be logged
Argument	Description								
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)								
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)								
<i>text</i>	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	<i>application.instancename.of_CreateNotifyText (subject, text)</i>										
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>application</i></td><td>Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)</td></tr> <tr> <td><i>subject</i></td><td>String containing text used in the mail messages subject line</td></tr> <tr> <td><i>text</i></td><td>String containing message text</td></tr> </tbody> </table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)	<i>subject</i>	String containing text used in the mail messages subject line	<i>text</i>	String containing message text
Argument	Description										
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)										
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)										
<i>subject</i>	String containing text used in the mail messages subject line										
<i>text</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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

applicationinstancename.of_GetLogFileStyle ()

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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	<i>application.instancename.of_GetLogSeverity()</i>
Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>mailsession</i>	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 `applicationinstancename.of_GetNotifyPreTitle()`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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 `applicationinstancename.of_GetNotifySeverity()`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>users</i>	String array into which the function places the list of users to be notified when errors occur (passed by reference)
<i>addresses</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>source</i>	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 *applicationinstancename.of_GetPredefinedSourceType ()*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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	<i>application.instancename.of_GetStyle()</i>						
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>application</i></td><td>Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)</td></tr><tr><td><i>instancename</i></td><td>Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)</td></tr></tbody></table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
Argument	Description						
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)						
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)						
Return value	<p>Integer. Returns the message box style:</p> <ul style="list-style-type: none">• DEFAULT or 0 PowerBuilder MessageBox• PFCWINDOW or 1 PFC w_message window						

Examples

```
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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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	<code>applicationinstancename.of_GetUnattended()</code>
Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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	<code>applicationinstancename.of.GetUser()</code>
Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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 *applicationinstancename.of_LoadPredefinedMsg ()*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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.

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

applicationinstancename.
 {, overridetitle } }

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)

Argument	Description
<i>messageid</i>	String specifying the identifier of the message to be displayed. This value must be in the msgnumber column of the ids_messages DataStore
<i>substitutions</i> (optional)	String array specifying substitution parameters for the message. The message source indicates substitution parameters with %s
<i>overridetitle</i> (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	<p>Call this function to display a predefined message from the ids_messages DataStore.</p> <p>This function displays either the PowerBuilder MessageBox or the PFC w_message window, depending on what you specify for the of_SetStyle function.</p> <p>The function uses substitution parameters to replace percent sign characters (%) in the predefined message. You define one element in the <i>substitution</i> 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.</p> <p>Use Syntax 2 to display w_message passing message text in a string.</p>
Examples	<p>This example calls the of_Message function:</p> <pre>gnv_app.inv_error.of_Message("EIS0210")</pre>

Syntax 2**Display a message using a passed string**

Description Displays message text and display information from passed arguments.

Access Public

Syntax *applicationinstancename.of_Message (title, message {, icon {, button {, default {, severity, print, userinput } } } })*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>title</i>	String specifying the title of the w_message dialog box
<i>message</i>	String specifying the message text

Argument	Description
<i>icon</i> (optional)	Icon enumerated data type indicating the icon you want to display on the left side of the message box. Values are: <ul style="list-style-type: none">• Information! (Default)• StopSign!• Exclamation!• Question!• None!
<i>button</i> (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: <ul style="list-style-type: none">• OK! (default)• OKCancel!• YesNo!• YesNoCancel!• RetryCancel!• AbortRetryIgnore!
<i>default</i> (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
<i>severity</i> (optional)	Integer indicating message severity
<i>print</i> (optional)	Boolean indicating whether to display a print button
<i>userinput</i> (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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>parms</i>	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
<i>instancename</i>	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 *applicationinstancename.of_SetBeep (boolean)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>boolean</i>	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	<code>application.instancename.of_SetLogFile (logfile)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>application</i></td><td>Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)</td></tr> <tr> <td><i>logfile</i></td><td>String specifying the name of the file to contain logged messages</td></tr> </tbody> </table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)	<i>logfile</i>	String specifying the name of the file to contain logged messages
Argument	Description								
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)								
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)								
<i>logfile</i>	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 <i>logfile</i> 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: <code>gnv_app.inv_error.of_SetLogFile & ("c:\eisapp\veislog.txt")</code>								
of_SetLogFileStyle									
Description	Specifies the formatting used for message logging.								
Access	Public								
Syntax	<code>application.instancename.of_SetLogFileStyle (logstyle)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>application</i></td><td>Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)</td></tr> <tr> <td><i>logstyle</i></td><td>Integer or n_cst_error constant specifying the log file style: <ul style="list-style-type: none"> • 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 </td></tr> </tbody> </table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)	<i>logstyle</i>	Integer or n_cst_error constant specifying the log file style: <ul style="list-style-type: none"> • 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
Argument	Description								
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)								
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this value is inv_error)								
<i>logstyle</i>	Integer or n_cst_error constant specifying the log file style: <ul style="list-style-type: none"> • 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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>level</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>mailsession</i>	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 <i>mailsession</i> 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 *applicationinstancename.of_SetNotifyPreTitle (prefix)*

Argument	Description
<i>application</i>	Global variable of type <i>n_cst_appmanager</i> or a customized <i>n_cst_appmanager</i> descendant (must be <i>gnv_app</i>)
<i>instancename</i>	Instance name of <i>n_cst_error</i> (the <i>n_cst_appmanager</i> default for this value is <i>inv_error</i>)
<i>prefix</i>	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 *applicationinstancename.of_SetNotifySeverity (threshold)*

Argument	Description
<i>application</i>	Global variable of type <i>n_cst_appmanager</i> or a customized <i>n_cst_appmanager</i> descendant. (must be <i>gnv_app</i>).

Argument	Description
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>threshold</i>	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:
	<pre>this.of_SetError(TRUE) this.inv_error.of_SetPredefinedSource & ("c:\eisapp\eiserr.txt") this.inv_error.of_SetNotifySeverity(4)</pre>

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
<i>application</i> .instancename.of_SetNotifyWho (<i>userlist</i> {, <i>addresslist</i> })	
Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>userlist</i>	String array containing the e-mail IDs of users to be notified when certain errors occur
<i>addresslist</i>	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

applicationinstancename.of_SetPredefinedSource (source)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>source</i>	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 `applicationinstancename.of_SetStyle (style)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>style</i>	Integer or n_cst_error constant specifying the message box style: <ul style="list-style-type: none"> • 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 `applicationinstancename.of_SetTimeOut (seconds)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>seconds</i>	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	<code>applicationinstancename.of_SetUnattended (boolean)</code>

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_error (the n_cst_appmanager default for this is inv_error)
<i>boolean</i>	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
<i>application</i>	Global variable of type n_cst_appmanager or a customized n_cst_appmanager descendant (must be gnv_app)
<i>instancename</i>	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())
```

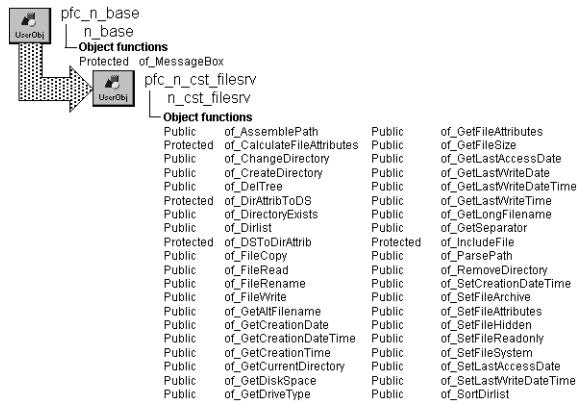
n_cst_filesrv

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.

Ancestry



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_filesvrwin32
See also	f_SetFilesvr n_cst_platform

Instance variables

N_cst_filesvr 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_filesvr includes precoded object functions, most of which are virtual functions with corresponding versions in n_cst_filesvr 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_filesvr for use with all n_cst_filesvr descendants.

N_cst_filesvr includes precoded object functions:

of_AssemblePath	of_GetFileAttributes
of_CalculateFileAttributes	of_GetFileSize
of_ChangeDirectory	of_GetLastAccessDate
of_CreateDirectory	of_GetLastWriteDate
of_DelTree	of_GetLastWriteDate
of_DirAttribToDS	of_GetLastWriteTime
of_DirectoryExists	of_GetLongFilename
of_DirList	of_GetSeparator
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_AssemblePath

Description Assembles a fully qualified directory path from its component parts.

Access Public

Syntax *instancename.of_AssemblePath (drive, path, filename {, extension })*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>drive</i>	String specifying the drive name. The colon is optional
<i>path</i>	String specifying the directory path. The initial backslash is optional
<i>filename</i>	String specifying the name of the file
<i>extension</i> (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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	String specifying the name of the files whose attributes will be set. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>readonly</i>	Boolean indicating the read-only attribute. If you specify NULL for this argument, the function does not change the read-only attribute
<i>hidden</i>	Boolean indicating the hidden attribute. If you specify NULL for this argument, the function does not change the hidden attribute
<i>system</i>	Boolean indicating the system attribute. If you specify NULL for this argument, the function does not change the system attribute
<i>archive</i>	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_filesvrwin32 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_filesvr for use with all n_cst_filesvr descendants.

Access

Public

Syntax

instancename.of_ChangeDirectory (directoryname)

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
<i>directoryname</i>	String

Return value	Integer. Always returns -1.
See also	n_cst_filesvwin32 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
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>
<i>directoryname</i>	String

Return value Integer. Always returns -1.

See also [n_cst_filesvwin32 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
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>
<i>directoryname</i>	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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>datastore</i>	N_ds
<i>entry</i>	N_cst_dirattrib
<i>file</i>	String
<i>group</i>	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	<i>instancename.of_DirectoryExists (directoryname)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>directoryname</i>	String
Return value	Boolean. Always returns FALSE.
See also	n_cst_filesvrwin32 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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filespec</i>	String
<i>filetype</i>	Long
<i>dirlist</i>	N_cst_dirlistattrib structure object (passed by reference)
<i>sorttype</i>	Integer

Return value Long. Always returns -1.

See also [n_cst_filesvrwin32 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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>datastore</i>	N_ds
<i>entry</i>	N_cst_dirattrib (passed by reference)
<i>row</i>	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_dirent,
        li_cnt)
        anv_dirlist[li_pos] = lnv_dirent
        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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>sourcefile</i>	String specifying the name of the file to be copied. If <i>sourcefile</i> does not specify a fully qualified path, the function works relative to the current directory
<i>targetfile</i>	String specifying the name of the target file. If <i>targetfile</i> does not specify a fully qualified path, the function works relative to the current directory
<i>append</i> (optional)	Boolean specifying whether to append to <i>targetfile</i> (TRUE) or replace it (FALSE)

Return value Integer. Returns values as follows:

- 1—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

Description

Access

Syntax

To read a file into a Blob variable

Reads a file into a Blob variable.

Public

instancename.of_FileRead (*filename*, *blob*)

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String specifying the name of the file to be read. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>blob</i>	Blob variable into which the function copies the contents of <i>filename</i> (passed by reference)

Return value	Long. Returns the size of <i>blob</i> 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.
Examples	Use Syntax 2 to read a file into a string array. This example calls the <code>of_FileRead</code> function (it assumes an <code>inv_filesv</code> instance variable):

```

Long    ll_return
Blob    llblb_file

ll_return = inv_filesv.of_FileRead &
            (sle_filename.text, llblb_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
<code>instancename</code>	Instance name of <code>n_cst_filesv</code>
<code>filename</code>	String specifying the name of the file to be read. If <code>filename</code> does not specify a fully qualified path, the function works relative to the current directory
<code>text</code>	String array into which the function copies the contents of <code>filename</code>

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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>oldname</i>	String
<i>newname</i>	String

Return value

Integer. Always returns -1.

See also

n_cst_filesvrwin32 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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	String specifying the name of the file to be written. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>blob</i>	Blob variable from which the function copies to <i>filename</i>
<i>append</i> (optional)	Boolean specifying whether to append to (TRUE) or overwrite (FALSE) <i>filename</i> . 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_filesvr has been created using the inv_filesvr instance variable:

```

Blob    lblb_file
... // Logic to populate lblb_file
IF inv_filesvr.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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	String specifying the name of the file to be written. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>text</i>	String from which the function copies to <i>filename</i>
<i>append</i> (optional)	Boolean specifying whether to append to (TRUE) or overwrite (FALSE) <i>filename</i> . 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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String

Return value

String. Always returns an empty string.

See also

n_cst_filesvrwin32 of_GetAltFilename function

of_GetCreationDate**Description**

Retrieves the date the file was created.

Access

Public

Syntax

instancename.of_GetCreationDate (*filename*, *date*)

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String specifying the name of the file whose creation date is to be returned. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>date</i>	Date variable into which the function places the date <i>filename</i> 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.

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
<code>instancename</code>	Instance name of <code>n_cst_filesrv</code>
<code>filename</code>	String
<code>date</code>	Date
<code>time</code>	Time

Return value Integer. Always returns -1.

See also `n_cst_filesrw32` `of_GetCreationTime` function

of_GetCreationTime

Description Retrieves the time the file was created.

Access Public

Syntax `instancename.of_GetCreationTime (filename, time)`

Argument	Description
<code>instancename</code>	Instance name of <code>n_cst_filesrv</code>
<code>filename</code>	String specifying the name of the file whose creation time is to be returned. If <code>filename</code> does not specify a fully qualified path, the function works relative to the current directory
<code>time</code>	Time variable into which the function places the time <code>filename</code> 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_GetCreationDate function.

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
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>

Return value

String. Always returns an empty string.

See also

n_cst_filesvrwin32 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)

instancename.of_GetDiskSpaceEx (drive, total, free)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>
<i>drive</i>	Character
<i>total</i>	Long (of_GetDiskSpace syntax) or decimal (of_GetDiskSpaceEx syntax)
<i>free</i>	Long (of_GetDiskSpace syntax) or decimal (of_GetDiskSpaceEx syntax)

Return value	Integer. Always returns -1.
See also	n_cst_filesrwin32 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_filesr for use with all n_cst_filesr descendants.

Access Public

Syntax *instancename.of_GetDriveType (drive)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesr
<i>drive</i>	Character

Return value Integer. Always returns -1.

See also n_cst_filesrwin32 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_filesr for use with all n_cst_filesr descendants.

Access Public

Syntax *instancename.of_GetFileAttributes (filename, readonly, hidden, system, subdirectory, archive)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesr
<i>filename</i>	String
<i>readonly</i>	Boolean
<i>hidden</i>	Boolean
<i>system</i>	Boolean
<i>subdirectory</i>	Boolean
<i>archive</i>	Boolean

Return value Integer. Always returns -1.

See also n_cst_filesrwin32 of _GetFileAttributes function

of_GetFileSize

Description	Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type <i>n_cst_filesrv</i> for use with all <i>n_cst_filesrv</i> descendants.						
Access	Public						
Syntax	<i>instancename.of_GetFileSize (filename)</i>						
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of <i>n_cst_filesrv</i></td></tr><tr><td><i>filename</i></td><td>String</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>	<i>filename</i>	String
Argument	Description						
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>						
<i>filename</i>	String						
Return value	Double. Always returns -1.						
See also	<i>n_cst_filesvrwin32</i> <i>of_GetFileSize</i> function						

of_GetLastAccessDate

Description	Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type <i>n_cst_filesrv</i> for use with all <i>n_cst_filesrv</i> descendants.								
Access	Public								
Syntax	<i>instancename.of_GetLastAccessDate (filename, date)</i>								
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of <i>n_cst_filesrv</i></td></tr><tr><td><i>filename</i></td><td>String specifying the name of the file whose last access date is to be returned. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory</td></tr><tr><td><i>date</i></td><td>Date variable into which the function places the date <i>filename</i> last accessed (passed by reference)</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>	<i>filename</i>	String specifying the name of the file whose last access date is to be returned. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory	<i>date</i>	Date variable into which the function places the date <i>filename</i> last accessed (passed by reference)
Argument	Description								
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>								
<i>filename</i>	String specifying the name of the file whose last access date is to be returned. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory								
<i>date</i>	Date variable into which the function places the date <i>filename</i> last accessed (passed by reference)								
Return value	Integer. Always returns -1.								
See also	<i>n_cst_filesvrwin32</i> <i>of_GetLastAccessDate</i> function								

of_GetLastWriteDate

Description	Retrieves the date the file was last modified.
Access	Public
Syntax	<i>instancename.of_GetLastWriteDate (filename, date)</i>

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String specifying the name of the file whose last write date is to be returned. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>date</i>	Date variable into which the function places the date <i>filename</i> 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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String
<i>date</i>	Date
<i>time</i>	Time

Return value Integer. Always returns -1.

See also n_cst_filesrwin32 of_GetLastWriteDateTime function

of_GetLastWriteTime

Description Retrieves the time the file was last modified.

Access Public

Syntax *instancename.of_GetLastWriteTime (filename, time)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String specifying the name of the file whose last write date is to be returned. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>time</i>	Time variable into which the function place the time <i>filename</i> 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_GetLastWriteDate 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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String

Return value String. Always returns an empty string.

See also n_cst_filesvrwin32 of_GetLongFilename function

of_GetSeparator

Description Retrieves the directory separator character.

Access	Public				
Syntax	<code>instancename.of_GetSeparator()</code>				
	<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th><th style="text-align: center;">Description</th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>instancename</i></td><td>Instance name of n_cst_filesrv</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_filesrv
Argument	Description				
<i>instancename</i>	Instance name of n_cst_filesrv				
Return value	String. Returns the <i>is_separator</i> instance variable.				
Usage	N_cst_filesrv descendants can call this function to create fully qualified pathnames.				
Examples	<p>This example calls the <i>of_GetSeparator</i> function:</p> <pre>String is_separator is_separator = inv_filesrv.of_GetSeparator()</pre>				

of_IncludeFile

Description	Called by the <i>of_DirList</i> function to report whether a file contains the specified attribute settings.										
Access	Protected										
Syntax	<code>instancename.of_IncludeFile(filename, attributemask, fileattrib)</code>										
	<table border="1"> <thead> <tr> <th style="text-align: center;">Argument</th><th style="text-align: center;">Description</th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>instancename</i></td><td>Instance name of n_cst_filesrv</td></tr> <tr> <td style="text-align: center;"><i>filename</i></td><td>String specifying the file to be accessed. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory</td></tr> <tr> <td style="text-align: center;"><i>attributemask</i></td><td>Long containing the bit string that determines whether <i>filename</i> is included by <i>of_DirList</i></td></tr> <tr> <td style="text-align: center;"><i>fileattrib</i></td><td>UnsignedLong containing the file's attribute bits</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_filesrv	<i>filename</i>	String specifying the file to be accessed. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory	<i>attributemask</i>	Long containing the bit string that determines whether <i>filename</i> is included by <i>of_DirList</i>	<i>fileattrib</i>	UnsignedLong containing the file's attribute bits
Argument	Description										
<i>instancename</i>	Instance name of n_cst_filesrv										
<i>filename</i>	String specifying the file to be accessed. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory										
<i>attributemask</i>	Long containing the bit string that determines whether <i>filename</i> is included by <i>of_DirList</i>										
<i>fileattrib</i>	UnsignedLong containing the file's attribute bits										
Return value	Boolean. Returns TRUE if <i>filename</i> should be included and FALSE if it should not.										
Usage	The <i>of_DirList</i> function calls this function to determine whether to include the file in its list of files.										
Examples	<p>This example is from the <i>of_DirList</i> function:</p> <pre>... DO WHILE li_RC = 0 lui_Attrib = Asc(lstr_FindData.c_Attrib) // Determine if this file should be included.</pre>										

```

        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
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>
<i>path</i>	String specifying the fully qualified filename
<i>drive</i>	String into which the function places the drive specification, including the colon and backslash (if specified in <i>fullname</i>) (passed by reference)
<i>directorypath</i>	String into which the function places the directory path (passed by reference)
<i>filename</i>	String into which the function places the filename without the extension (passed by reference)
<i>extension</i> (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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>directoryname</i>	String

Return value Integer. Always returns -1.

See also n_cst_filesvrwin32 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_filesvr for use with all n_cst_filesvr descendants.

Access Public

Syntax *instancename.of_SetCreationDateTime (filename, date, time)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	String
<i>date</i>	Date
<i>time</i>	Time

Return value Integer. Always returns -1.

See also n_cst_filesvrwin32 of_SetCreationDateTime function

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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	String specifying the name of the file whose archive attribute is to be set. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>archive</i>	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 <i>inv_filesrv</i> 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 <i>n_cst_filesrv</i> for use with all <i>n_cst_filesrv</i> descendants.
Access	Public
Syntax	<i>instancename.of_SetFileAttributes (filename, readonly, hidden, system, archive)</i>

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>
<i>filename</i>	String
<i>readonly</i>	Boolean
<i>hidden</i>	Boolean
<i>system</i>	Boolean
<i>archive</i>	Boolean

Return value	Integer. Always returns -1.
See also	<i>n_cst_filesrwin32</i> of_SetFileAttributes function

of_SetFileHidden

Description	Calls the of_SetFileAttributes function to set a file's hidden attribute.
Access	Public
Syntax	<i>instancename.of_SetFileHidden (filename, hidden)</i>

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>
<i>filename</i>	String specifying the name of the file whose hidden attribute is to be set. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>hidden</i>	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	<i>instancename.of_SetFileReadonly (filename, readonly)</i>

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String specifying the name of the file whose read-only attribute is to be set. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>readonly</i>	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	<i>instancename.of_SetFileSystem (filename, system)</i>

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	String specifying the name of the file whose system attribute is to be set. If <i>filename</i> does not specify a fully qualified path, the function works relative to the current directory
<i>system</i>	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 <i>inv_filesrv</i> 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 <i>n_cst_filesrv</i> for use with all <i>n_cst_filesrv</i> descendants.								
Access	Public								
Syntax	<i>instancename.of_SetLastAccessDate (filename, date)</i>								
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of <i>n_cst_filesrv</i></td></tr><tr><td><i>filename</i></td><td>String</td></tr><tr><td><i>date</i></td><td>Date</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>	<i>filename</i>	String	<i>date</i>	Date
Argument	Description								
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>								
<i>filename</i>	String								
<i>date</i>	Date								
Return value	Integer. Always returns -1.								
See also	<i>n_cst_filesrw32</i> <i>of_SetLastAccessDate</i> function								

of_SetLastWriteDateTime

Description	Virtual function only. By defining virtual functions in this object, you can declare a reference variable of type <i>n_cst_filesrv</i> for use with all <i>n_cst_filesrv</i> descendants.										
Access	Public										
Syntax	<i>instancename.of_SetLastWriteDateTime (filename, date, time)</i>										
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of <i>n_cst_filesrv</i></td></tr><tr><td><i>filename</i></td><td>String</td></tr><tr><td><i>date</i></td><td>Date</td></tr><tr><td><i>time</i></td><td>Time</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>	<i>filename</i>	String	<i>date</i>	Date	<i>time</i>	Time
Argument	Description										
<i>instancename</i>	Instance name of <i>n_cst_filesrv</i>										
<i>filename</i>	String										
<i>date</i>	Date										
<i>time</i>	Time										
Return value	Integer. Always returns -1.										
See also	<i>n_cst_filesrw32</i> <i>of_SetLastWriteDateTime</i> 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
<i>instancename</i>	Instance name of n_cst_filesv
<i>dirlist</i>	Array of type n_cst_dirattrib containing the output from the of_DirList function (passed by reference)
<i>sorttype</i>	Integer specifying how to sort <i>dirlist</i> : <ul style="list-style-type: none"> • 1 By filename • 2 By extension • 3 By last write date/time • 4 By filesize
<i>ascending</i> (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_filesv.of_DirList &
            (ls_currdir, 0, lnv_dirlist)
...
inv_filesv.of_SortDirList(lnv_dirlist, 1, TRUE)
...

```

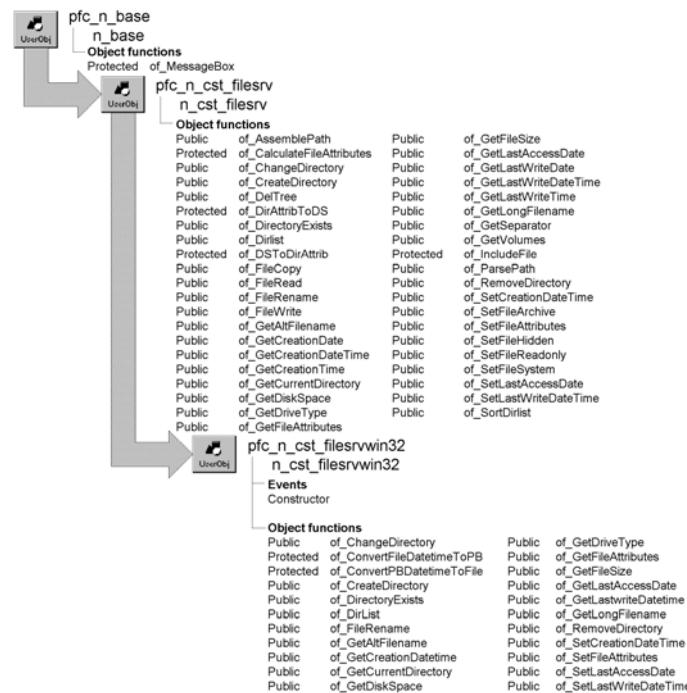
n_cst_filesvrwin32

Description

Platform-specific file management functions for applications running under Windows. Platform-specific file management functions require a call to a platform-specific external function. When you call the f_SetFilesvr 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_SetFilesr 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_filesr) 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
	b_inherit	Boolean
os_securityattributes	ch_description	Character
	ul_length	UnsignedLong
	ch_alternatefilename[14]	Character
os_finddata	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
os_fileopeninfo	ul_reserved1	UnsignedLong
	c_fixed_disk	Character
	c_length	Character
	c.pathname[128]	Character
	ui_dos_error	UnsignedInteger

Structure	Field	Data type
	ui_na1	UnsignedInteger
	ui_na2	UnsignedInteger

See also

[f_SetFilesrv](#)
[n_cst_filesrv](#)
[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:

<code>of_ChangeDirectory</code>	<code>of_GetDriveType</code>
<code>of_ConvertFileDateToPB</code>	<code>of_GetFileAttributes</code>
<code>of_ConvertPBDTimeToFile</code>	<code>of_GetFileSize</code>
<code>of_CreateDirectory</code>	<code>of_GetLastAccessDate</code>
<code>of_DirectoryExists</code>	<code>of_GetLastWriteDate</code>
<code>of_DirList</code>	<code>of_GetLongFilename</code>
<code>of_FileRename</code>	<code>of_RemoveDirectory</code>
<code>of_GetAltFilename</code>	<code>of_SetCreationDate</code>
<code>of_GetCreationDate</code>	<code>of_SetFileAttributes</code>
<code>of_GetCurrentDirectory</code>	<code>of_SetLastAccessDate</code>
<code>of_GetDiskSpace</code>	<code>of_SetLastWriteDate</code>

of_ChangeDirectory

Description	Changes the current working directory.
Access	Public
Syntax	<code>instancename.of_ChangeDirectory (directoryname)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_filesv
<i>directoryname</i>	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_filesv instance variable): <pre>Integer li_return String ls_language ls_language = sle_curr_language.Text li_return = inv_filesv.of_ChangeDirectory & ("\\realapp") IF li_return = 1 THEN SetProfileString("realapp.ini", "language", & "currentlanguage", ls_language) END IF</pre>

of_ConvertFileDialogTimeToPB

Description	Converts a file date (UnsignedInteger) into a PowerBuilder date.
Access	Protected
Syntax	<code>instancename.of_ConvertFileDialogTimeToPB (filedatetime, date, time)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_filesv

Argument	Description
<i>filedatetime</i>	Structure of type os_filedatetime containing a file's system date and time
<i>date</i>	Date variable into which the function places the date (passed by reference)
<i>time</i>	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_ConvertPBDateTimeToFile &
    (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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>date</i>	Date variable containing a date to be converted
<i>time</i>	Time variable containing a time to be converted
<i>datetime</i>	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_SetLastWriteDate 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
<i>instancename</i>	Instance name of n_cst_filesv
<i>directoryname</i>	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_filesv instance variable):

```
IF inv_filesv.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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>directoryname</i>	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_filesvr instance variable):

```
Integer li_return

IF NOT inv_filesvr.of_DirectoryExists &
    ("c:\rkndoc\temp") THEN
    li_return = inv_filesvr.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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filespec</i>	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

Argument	Description
<i>filetype</i>	Long specifying one or more types of files to return. Types are: <ul style="list-style-type: none"> • 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
<i>dirlist</i>	Array of type n_cst_dirattrib into which the function places information for the requested files (passed by reference)

Return value	Long. Returns the number of elements in the <i>dirlist</i> array if the function succeeds and -1 if an error occurs.
Usage	To further sort the <i>dirlist</i> array, use the of_SortDirList function. To list several types, add the <i>filetype</i> values together. For example, to list read/write files, subdirectories, and drives, specify 0+16+16384 or 16400 for <i>filetype</i> .
Examples	This function calls the FindNextFileA external function. This example calls the of_DirList function (this example assumes an <i>inv_filesrv</i> 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	<i>instancename.of_FileRename (oldname, newname)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
<i>oldname</i>	String specifying the complete path of the file to be renamed
<i>newname</i>	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_filesvr.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)
ENDIF
```

of_GetAltFilename

Description Retrieves the internal 8.3 alternate filename for a passed long filename.

Access Public

Syntax *instancename.of_GetAltFilename (longfilename)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
<i>longfilename</i>	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_filesvr instance variable):

```

String    ls_filename, ls_return

ls_filename = sle_source.text
ls_return = inv_filesvr.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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	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
<i>date</i>	Date into which the function places the creation date (passed by reference)
<i>time</i>	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_filesvr instance variable):

```

Integer  li_return
String   ls_filename
Date    ldt_date
Time    lt_time

ls_filename = sle_source.text
li_return = inv_filesvr.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	<i>instancename.of_GetCurrentDirectory ()</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
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_filesvr instance variable):

```
String    ls_return

ls_return = inv_filesvr.of_GetCurrentDirectory()
IF ls_return <> is_app_dir THEN
    inv_filesvr.of_ChangeDirectory(is_app_dir)
END IF
```

of_GetDiskSpace

Description	Retrieves space information for a specified disk drive.
Access	Public
Syntax	<i>instancename.of_GetDiskSpace (drive, total, free)</i> <i>instancename.of_GetDiskSpaceEx (drive, total, free)</i>

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
<i>drive</i>	Character specifying the drive letter
<i>total</i>	Long (of_GetDiskSpace syntax) or decimal (of_GetDiskSpaceEx syntax) into which the function returns the total bytes on <i>drive</i> , passed by reference
<i>free</i>	Long (of_GetDiskSpace syntax) or decimal (of_GetDiskSpaceEx syntax) into which the function returns the free space on <i>drive</i> , 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.

It is best to use the of_GetDiskSpaceEx syntax. The of_GetDiskSpace syntax is supported but obsolete. It returns a negative number when the calculated amount of free disk space exceeds 2 gigabytes.

Examples This example calls the of_GetDiskSpace function:

```

Character   lc_drive
Decimal    ldec_total, ldec_free
Integer     li_return

lc_drive = lb_drive.SelectedItem()
li_return = inv_filesvr.of_GetDiskSpaceEx &
            (lc_drive, ldec_total, ldec_free)
IF li_return = 1 THEN
    MessageBox("File Service", "Drive " + lc_drive &
+ "~r~n" +String(ldec_total)+" total bytes~r~n" &
+ String(ldec_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
<i>instancename</i>	Instance name of n_cst_filesvr

Argument	Description
<i>drive</i>	Character specifying the drive letter
Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none">0 Undetermined drive type1 Root directory does not exist2 Removable disk3 Nonremovable disk4 Remote (network) drive5 CD-ROM drive6 RAM disk
Examples	<p>This example calls the <code>of_GetDriveType</code> function (this example assumes an <code>inv_filesrv</code> instance variable):</p> <pre>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])</pre>

of_GetFileAttributes

Description	Retrieves all attributes for a specified file.
Access	Public
Syntax	<code>instancename.of_GetFileAttributes (filename, readonly, hidden, system, subdirectory, archive)</code>

Argument	Description
<i>instancename</i>	Instance name of <code>n_cst_filesrv</code>
<i>filename</i>	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
<i>readonly</i>	Boolean indicating whether <i>filename</i> is read-only (passed by reference)

Argument	Description
<i>hidden</i>	Boolean indicating whether <i>filename</i> is hidden (passed by reference)
<i>system</i>	Boolean indicating whether <i>filename</i> is a system file (passed by reference)
<i>subdirectory</i>	Boolean indicating whether <i>filename</i> is a subdirectory (passed by reference)
<i>archive</i>	Boolean indicating whether <i>filename</i> 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_filesvr` instance variable):

```

Integer li_return
String ls_filename

ls_filename = sle_source.text
li_return = inv_filesvr.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
<i>instancename</i>	Instance name of <code>n_cst_filesvr</code>
<i>filename</i>	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 |
|---------------------|--|
| <i>instancename</i> | Instance name of <i>n_cst_filesrv</i> |
| <i>filename</i> | 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 |
| <i>date</i> | 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
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	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
<i>date</i>	Date into which the function places the last write date (passed by reference)
<i>time</i>	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_filesvr instance variable):

```

Integer li_return
String ls_filename
Date ldt_date
Time lt_time

ls_filename = sle_source.text
li_return = inv_filesvr.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
<i>instancename</i>	Instance name of <i>n_cst_filesvr</i>
<i>shortfilename</i>	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_filesvr* instance variable):

```
String    ls_filename, ls_return

ls_filename = sle_source.text
ls_return = inv_filesvr.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	<i>instancename.of_RemoveDirectory (directoryname)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvwin
<i>directoryname</i>	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_filesv 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_filesv.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
<i>instancename</i>	Instance name of n_cst_filesrv
<i>filename</i>	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
<i>date</i>	Date variable containing the new creation date
<i>time</i>	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	<code>instancename.of_SetFileAttributes (filename, readonly, hidden, system, archive)</code>
Argument	Description
<code>instancename</code>	Instance name of n_cst_filesrwin32
<code>filename</code>	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
<code>readonly</code>	Boolean indicating whether <code>filename</code> is read-only. If you specify NULL, the function doesn't modify this attribute
<code>hidden</code>	Boolean indicating whether <code>filename</code> is hidden. If you specify NULL, the function doesn't modify this attribute
<code>system</code>	Boolean indicating whether <code>filename</code> is a system file. If you specify NULL, the function doesn't modify this attribute
<code>archive</code>	Boolean indicating whether <code>filename</code> 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_filesr instance variable):

```

Integer li_return
String ls_filename

ls_filename = sle_source.text
li_return = inv_filesr.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	<i>instancename.of_SetLastAccessDate (filename, date)</i>								
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of n_cst_filesrv</td></tr><tr><td><i>filename</i></td><td>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</td></tr><tr><td><i>date</i></td><td>Date variable containing the new last-accessed date</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_filesrv	<i>filename</i>	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	<i>date</i>	Date variable containing the new last-accessed date
Argument	Description								
<i>instancename</i>	Instance name of n_cst_filesrv								
<i>filename</i>	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								
<i>date</i>	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_filesr instance variable): <pre>Integer li_return String ls_filename Date ldt_date ls_filename = sle_source.text ldt_date = Today() li_return = inv_filesr.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</pre>								

of_SetLastWriteDateTime

Description Sets a file's last-modified date and time.

Access Public

Syntax *instancename.of_SetLastWriteDateTime (filename, date, time)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_filesvr
<i>filename</i>	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
<i>date</i>	Date variable containing the new last-modified date
<i>time</i>	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_filesvr instance variable):

```

Integer li_return
String ls_filename
Date ldt_date
Time lt_time

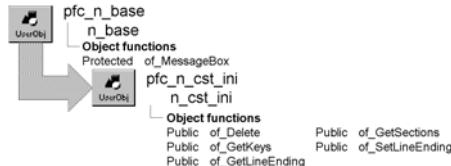
ls_filename = sle_source.text
ldt_date = Today()
lt_time = Now()
li_return = inv_filesvr.of_SetLastWriteDateTime &
            (ls_filename, ldt_date, lt_time)
IF li_return = -1 THEN
    MessageBox("File Service", &
               "Error modifying " + ls_filename)
ELSE
    MessageBox("File Service", 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 pre-coded event:

Constructor

Constructor

Description	Initializes the <code>is_lineend</code> 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:

<code>of_Delete</code>	<code>of_GetSections</code>
<code>of_GetKeys</code>	<code>of_SetLineEnding</code>
<code>of_GetLineEnding</code>	

`of_Delete`

Description Removes the specified entry or section from an INI file.

Access Public

Syntax `instancename.of_Delete (file, section {, key })`

Argument	Description
<code>instancename</code>	Instance name of <code>n_cst_inifile</code>
<code>file</code>	String naming the INI file containing the entry or section to be deleted
<code>section</code>	String naming the section to be deleted. If you also specify <code>key</code> , this value names the section containing the entry to be deleted
<code>key</code> (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
<i>instancename</i>	Instance name of <i>n_cst_inifile</i>
<i>file</i>	String naming the INI file containing the section whose keys are retrieved
<i>sectionname</i>	String containing the name of the section whose keys are retrieved
<i>keynames</i>	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
<i>instancename</i>	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"
...
CASE "~r~n"
...
```

of_GetSections

Description Retrieves all sections from an INI file.

Access Public

Syntax *instancename.of_GetSections (file, sectionnames)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_inifile
<i>file</i>	String naming the INI file whose sections are retrieved
<i>sectionnames</i>	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	<i>instancename.of_SetLineEnding (character)</i>
--------	--

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_inifile</i>
<i>character</i>	String containing the line end character or characters: • ~r~n Carriage return - newline (Windows)

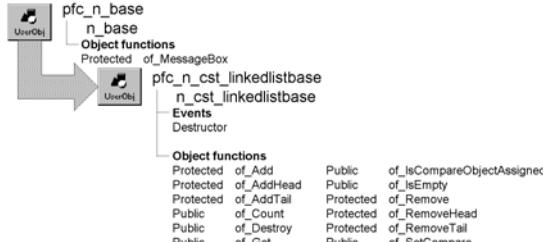
Return value	Integer. Returns 1 if the function succeeds -1 if an error occurs.
--------------	--

Examples	This example is from the <i>n_cst_inifile</i> Constructor event:
----------	--

```
Environment lenv_obj

GetEnvironment (lenv_obj)
if IsValid (lenv_obj) then
    choose case lenv_obj.ostype
        case aix!, osfl!, hpx!, aix!
            of_SetLineEnding("~n")
        ...
        case else
            of_SetLineEnding("~r~n")
        end choose
    else
        of_SetLineEnding ("~r~n")
    end if
```

n_cst_linkedListbase

Description	List service ancestor object. Implement linked list processing through one of this object's descendants:																								
	<ul style="list-style-type: none"> • N_cst_list • N_cst_queue • N_cst_stack 																								
Ancestry	 <pre> graph TD pfc_n_base[pfc_n_base] --> n_base[n_base] n_base --> ObjectFunctions[Object functions] ObjectFunctions --> of_MessageBox[of_MessageBox] of_MessageBox -- Protected --> pfc_n_cst_linkedListbase[pfc_n_cst_linkedListbase] pfc_n_cst_linkedListbase --> n_cst_linkedListbase[n_cst_linkedListbase] n_cst_linkedListbase -- Events --> Destructor[Destructor] </pre> <p>Object functions</p> <table border="0"> <tr> <td>Protected</td> <td>of_Add</td> <td>Public</td> <td>of_IsCompareObjectAssigned</td> </tr> <tr> <td>Protected</td> <td>of_AddHead</td> <td>Public</td> <td>of_IsEmpty</td> </tr> <tr> <td>Protected</td> <td>of_AddTail</td> <td>Protected</td> <td>of_Remove</td> </tr> <tr> <td>Public</td> <td>of_Count</td> <td>Protected</td> <td>of_RemoveHead</td> </tr> <tr> <td>Public</td> <td>of_Destroy</td> <td>Protected</td> <td>of_RemoveTail</td> </tr> <tr> <td>Public</td> <td>of_Get</td> <td>Public</td> <td>of_SetCompare</td> </tr> </table>	Protected	of_Add	Public	of_IsCompareObjectAssigned	Protected	of_AddHead	Public	of_IsEmpty	Protected	of_AddTail	Protected	of_Remove	Public	of_Count	Protected	of_RemoveHead	Public	of_Destroy	Protected	of_RemoveTail	Public	of_Get	Public	of_SetCompare
Protected	of_Add	Public	of_IsCompareObjectAssigned																						
Protected	of_AddHead	Public	of_IsEmpty																						
Protected	of_AddTail	Protected	of_Remove																						
Public	of_Count	Protected	of_RemoveHead																						
Public	of_Destroy	Protected	of_RemoveTail																						
Public	of_Get	Public	of_SetCompare																						
Library	PFCAPSrv.PBL PFEAPSrv.PBL																								
Object relationships	n_cst_linkedListnode n_cst_linkedListnodecompare																								
Usage	PFC objects use this object as the ancestor for other list-processing objects.																								
Descendants	n_cst_list n_cst_queue n_cst_stack																								
See also	n_cst_tree																								

Instance variables

N_cst_linkedListbase includes instance variables:

Instance variable	Description	Data type	Access	Usage
inv_compare	Reference to comparison object	N_cst_linkedListnodecompare	Protected	Set with of_SetCompare (default is n_cst_linkedListnodecompare)
inv_head	First node in the list	N_cst_linkedListnode	Protected	Internal
inv_nil	Always nil	N_cst_linkedListnode	Protected	Internal
inv_tail	Last node in the list	N_cst_linkedListnode	Protected	Internal

Events

N_cst_linkedListbase includes a precoded event:

Destructor

Destructor

- Description Destroys existing nodes.
Usage This event executes when the object is destroyed.

Functions

N_cst_linkedListbase includes precoded object functions:

of_Add	of_IsCompareObjectAssigned
of_AddHead	of_IsEmpty
of_AddTail	of_Remove
of_Count	of_RemoveHead
of_Destroy	of_RemoveTail
of_Get	of_SetCompare

of_Add

- Description Adds a new node to the linked list.

Access	Protected
Syntax	<code>instancename.of_Add (newnode, where)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedlistbase
<i>newnode</i>	N_cst_linkedlistnode instance to be added to the linked list
<i>where</i>	N_cst_linkedlistnode instance after which <i>newnode</i> is added. This node must already be in the list
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Usage	Internal.
Examples	This example calls the of_Add function:

```
Return this.of_Add(inv_newnode, inv_currnode)
```

of_AddHead

Description Adds a new node to the beginning of the linked list.

Access Protected

Syntax `instancename.of_AddHead (newnode)`

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedlistbase
<i>newnode</i>	N_cst_linkedlistnode instance to be added at the beginning of the linked list

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

Usage Internal.

Examples This example is from the of_AddTail function:

```
...
if isnull(inv_head) or not isvalid(inv_head) then
    return of_AddHead(anv_newtailnode)
end if
...
```

of_AddTail

Description Adds a new node to the end of the linked list.

Access Protected

n_cst_linkedListbase

Syntax	<i>instancename.of_AddTail (newnode)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedListbase
<i>newnode</i>	N_cst_linkedListnode instance to be added at the end of the linked list

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

Usage Internal.

Examples This example is from the of_Add function:

```
...
If anv_insertafternode = inv_tail Then
    Return of_AddTail(anv_newnode)
End If
...
```

of_Count

Description Retrieves a count of the number of nodes in the linked list.

Access Public

Syntax *instancename.of_Count ()*

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedListbase

Return value Long. Returns the number of nodes in the linked list.

Examples This example calls the of_Count function:

```
...
ll_count = inv_list.of_Count()
st_result.Text = &
    "Item count: " + String(ll_count)
```

of_Destroy

Description Destroys all nodes in the linked list.

Access Public

Syntax *instancename.of_Destroy ()*

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedListbase

Return value Integer. Returns the number of nodes destroyed if the function succeeds and -1 if an error occurs.

Examples This example calls the of_Destroy function:

```
...li_count = inv_queue.of_Destroy()
st_result.Text = &
String(li_count) + ' items deleted'...
```

of_Get

Description Retrieves all nodes in a list.

Access Public

Syntax *instancename.of_Get (nodes)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedlistbase
<i>node</i>	N_cst_linkedlistnode array into which the function places references to the nodes in the list (passed by reference)

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

Examples This example calls the of_Get function:

```
Long ll_rc
n_cst_linkedlistnode inv_nodes[ ]

ll_rc = inv_list.of_Get(inv_nodes)
...
```

of_IsCompareObjectAssigned

Description Reports whether a comparison object has been assigned.

Access Public

Syntax *instancename.of_IsCompareObjectAssigned ()*

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedlistbase

Return value Boolean. Returns TRUE if a comparison object has been assigned and FALSE if no comparison object has been assigned.

Examples

This example calls the of_IsCompareObjectAssigned function:

```
IF NOT inv_list.of_IsCompareObjectAssigned() THEN
    MessageBox("Comparison object", "Not assigned")
ELSE
    MessageBox("Comparison object", "Is assigned")
END IF
```

of_IsEmpty

Description

Reports whether a list is empty.

Access

Public

Syntax

instancename.of_IsEmpty ()

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_linkedListbase</i>

Return value

Integer. Returns TRUE if the list is empty and FALSE if it is not.

Examples

This example calls the of_IsEmpty function:

```
IF NOT inv_list.of_IsEmpty() THEN
    MessageBox("List", "Is empty")
ELSE
    MessageBox("List", "Is not empty")
END IF
```

of_Remove

Description

Removes the specified node from a linked list, resetting pointers as appropriate.

Access

Protected

Syntax

instancename.of_Remove (*removenode*)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_linkedListbase</i>
<i>removenode</i>	<i>N_cst_linkedListnode</i> instance to remove from the linked list (passed by reference)

Return value

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

Usage

Internal.

of_RemoveHead

Description Removes the first node from a linked list, resetting pointers as appropriate and returning the removed node.

Access Protected

Syntax *instancename.of_RemoveHead (removenode)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedListbase
<i>removenode</i>	N_cst_linkedListnode instance into which the function places a reference to the returned node (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 n_cst_list of_RemoveHead function:

```
Return Super::of_RemoveHead(anv_headnode)
```

of_RemoveTail

Description Removes the last node from a linked list, resetting pointers as appropriate and returning the removed node.

Access Protected

Syntax *instancename.of_RemoveTail (removenode)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedListbase
<i>removenode</i>	N_cst_linkedListnode instance into which the function places a reference to the returned node (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 n_cst_list of_RemoveTail function:

```
Return Super::of_RemoveTail(anv_tailnode)
```

of_SetCompare

Description Establishes a comparison object other than n_cst_linkedListnodecompare.

Access Public

n_cst_linkedListbase

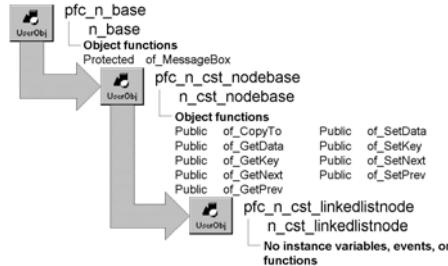
Syntax	<i>instancename</i> . of_SetCompare (<i>comparisonobject</i>)
Argument	Description
<i>instancename</i>	Instance name of n_cst_linkedListbase
<i>comparisonobject</i>	N_cst_linkedListnodecompare descendant instance that overrides the of_Compare function
Return value	Integer. Returns 1 if the function succeeds, -1 if an error occurs, and -2 if the comparison object has already been set.
Usage	Call this function to establish a customized comparison object. List service objects uses the comparison object's of_Compare function when handling sorted lists. The default comparison object (n_cst_linkedListnodecompare) can handle comparison for simple data types. But if your list's key items contain some other data type, you must create a customized n_cst_linkedListnodecompare descendant and implement an of_Compare function to handle sort order.
Examples	This example calls the of_SetCompare function (it assumes an inv_customcompare instance variable):

```
inv_customcompare = CREATE n_cst_customcompare  
  
inv_list.of_SetCompare(inv_customcompare)
```

n_cst_linkedListnode

Description List instance object.

Ancestry



Library PFCAPSRV.PBL

PFEAPSRV.PBL

Usage Use `n_cst_linkedListnode` as the list instance object for stacks, queues, sorted lists, and unsorted lists.

To use `n_cst_linkedListnode`:

1 Declare a variable for your list (an instance variable in this example):

```
n_cst_list inv_list
```

2 Declare variables for list nodes (local variables in this example):

```
n_cst_linkedListnode lnv_node
```

3 Create the node instance:

```
lnv_node = CREATE n_cst_linkedListnode
```

4 Access the node's key and data values as appropriate:

```
lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
```

5 Call list-processing functions as appropriate:

```
li_return = inv_list.of_Add(lnv_node)
```

See also

`n_cst_list`

`n_cst_queue`

`n_cst_stack`

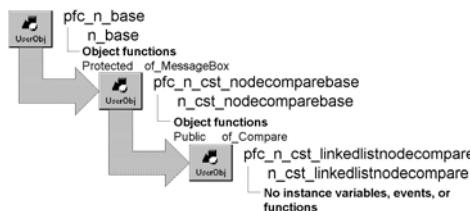
`n_cst_linkedListnodecompare`

n_cst_linkedListnodecompare

Description

Node comparison object. N_cst_list uses this object to compare values during find operations and before placement in a sorted list.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

This object can handle key comparison for simple data types but if your list's key items contain some other data type, you must create a customized *n_cst_linkedListnodecompare* descendant to handle comparison processing.

To use a customized descendant of *n_cst_linkedListnodecompare*:

- 1 Use the User Object painter to create a customized *n_cst_linkedListnodecompare* descendant.
- 2 In the customized *n_cst_linkedListnodecompare* descendant, implement a Public of_Cmpare function to compare key values in the two passed nodes. This function should take two arguments of type *n_cst_nodebase* (passed by value) and return integer values as follows:
 - **1** The key of the first node is less than the key of the second node
 - **0** The key of the first node is equal to the key of the second node
 - **2** The key of the first node is greater than the key of the second node

In this example, each passed node contains a reference to a custom class user object with state and last name instance variables to compare:

```
Any la_key1,la_key2
String ls_keytype1,ls_keytype2
n_cst_empinfo lInv_emp1, lInv_emp2

IF NOT IsValid(anv_node1) THEN
    Return -1
END IF
IF NOT IsValid(anv_node2) THEN
    Return -1
```

```
END IF
anv_node1.of_GetKey(la_key1)
IF IsNull(la_key1) THEN
    Return -1
END IF
anv_node2.of_GetKey(la_key2)
IF IsNull(la_key2) THEN
    Return -1
END IF

ls_keytype1 = ClassName(la_key1)
ls_keytype2 = ClassName(la_key2)

// Check class types
IF ls_keytype1 <> ls_keytype2 THEN
    Return -1
END IF

IF ls_keytype1 = "" THEN Return -1
IF IsNull(ls_keytype1) THEN Return -1
IF ls_keytype1 <> "n_cst_empinfo" THEN Return -1
IF ls_keytype2 = "" THEN Return -1
IF IsNull(ls_keytype2) THEN Return -1
IF ls_keytype2 <> "n_cst_empinfo" THEN Return -1
lnv_emp1 = la_key1 // Cast to n_cst_empinfo
lnv_emp2 = la_key2

// First compare State.
IF lnv_emp1.is_state < lnv_emp2.is_state THEN
    return this.LESSTHAN
elseif lnv_emp1.is_state > lnv_emp2.is_state THEN
    return this.GREATERTHAN
ELSE // States are equal; compare lname.
    IF lnv_emp1.is_lname < lnv_emp2.is_lname THEN
        Return this.LESSTHAN
    ELSEIF lnv_emp1.is_lname > lnv_emp2.is_lname THEN
        Return this.GREATERTHAN
    ELSE // State and lname are equal.
        Return this.EQUAL
    END IF
END IF
Return this.FAILURE
```

- 3 In the object that uses PFC list-processing, define an instance variable that uses your customized n_cst_linkedListnodecompare object as the data type:

```
n_cst_mycompare inv_customcompare
```

- 4 Create an instance of the customized comparison object and call the n_cst_list of_SetCompare function:

```
inv_customcompare = CREATE n_cst_mycompare  
inv_list.of_SetCompare(inv_customcompare)
```

- 5 Call list-processing functions as necessary (this example displays linked list contents in a ListBox):

```
n_cst_linkedListnode lnv_nodes[ ]  
Integer li_return, li_count  
Any la_data  
String ls_data  
n_cst_empinfo lnv_empinfo  
  
li_return = inv_list.of_Get(lnv_nodes)  
FOR li_count = 1 to li_return  
    lnv_nodes[li_count].of_GetData(la_data)  
    lnv_empinfo = la_data // Cast to n_cst_empinfo  
    ls_data = lnv_empinfo.is_lname &  
        + ", " + lnv_empinfo.is_fname  
    lb_1.AddItem(ls_data)  
NEXT
```

See also

[n_cst_list](#)
[n_cst_treenodecompare](#)

n_cst_list

Description	List processing object. You can implement either a sorted or unsorted list, optionally allowing duplicates.
Ancestry	<pre> graph TD pfc_n_base[pfc_n_base] --> n_base[n_base] n_base --> pfc_n_cst_linkedListbase[pfc_n_cst_linkedListbase] pfc_n_cst_linkedListbase --> pfc_n_cst_list[pfc_n_cst_list] pfc_n_cst_list --> n_cst_list[n_cst_list] </pre> <p>pfc_n_base</p> <ul style="list-style-type: none"> n_base Object functions Protected: of_MessageBox <p>pfc_n_cst_linkedListbase</p> <ul style="list-style-type: none"> n_cst_linkedListbase Events Destructor Object functions Protected: of_AddHead, of_AddTail, of_Count, of_Destroy, of_Get Public: of_IsCompareObjectAssigned, of_IsEmpty, of_Remove, of_RemoveHead, of_RemoveTail, of_SetCompare <p>pfc_n_cst_list</p> <ul style="list-style-type: none"> n_cst_list Object functions Public: of_Add, of_AddHead, of_AddTail, of_Find, of_IsDuplicatesAllowed, of_IsInList Protected: of_Remove, of_RemoveHead, of_RemoveTail Public: of_IsSorted, of_RemoveTail, of_SetDuplicatesAllowed, of_SetSorted
Library	PFCAPSRV.PBL PFEAPSRV.PBL
Object relationships	n_cst_linkedListnode n_cst_linkedListnodecompare
Usage	Use this object for simple linked lists and sorted linked lists. A sorted linked list can optionally allow duplicates.

To use this object:

1 Establish a variable of type n_cst_list:

```
n_cst_list inv_list
```

2 (Optional) Specify whether the list is sorted:

```
inv_list.of_SetSorted(TRUE)
```

3 (Optional) Specify whether a sorted list disallows duplicates:

```
inv_list.of_SetDuplicatesAllowed(FALSE)
```

4 Add nodes to the list:

```
n_cst_linkedlistnode lnv_node
Integer li_return

lnv_node = CREATE n_cst_linkedlistnode

lnv_node.of_SetKey(sle_key.text)
lnv_node.of_SetData(sle_data.text)
li_return = inv_list.of_Add(lnv_node)
```

5 Retrieve nodes from the list:

```
n_cst_linkedlistnode lnv_nodes[ ]
Integer li_return, li_count
Any la_data
String ls_data

lb_1.Reset()
li_return = inv_list.of_Get(lnv_nodes)
FOR li_count = 1 to li_return
    lnv_nodes[li_count].of_GetData(la_data)
    ls_data = String(la_data)
    lb_1.AddItem(ls_data)
NEXT
```

See also

[n_cst_linkedlistnode](#)
[n_cst_linkedlistnodecompare](#)
[n_cst_queue](#)
[n_cst_stack](#)
[n_cst_tree](#)

Instance variables

N_cst_list includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_duplicatesallowed	Indicates whether the list allows duplicates	Boolean	Protected	Set with of_SetDuplicatesAllowed (the default is TRUE)
ib_sorted	Indicates whether list is sorted	Boolean	Protected	Set with of_SetSorted (the default is FALSE)

Functions

N_cst_list includes precoded object functions:

of_Add	of_IsSorted
of_AddHead	of_Remove
of_AddTail	of_RemoveHead
of_Find	of_RemoveTail
of_IsDuplicatesAllowed	of_SetDuplicatesAllowed
of_IsInList	of_SetSorted

of_Add

Description	Adds a new node to the linked list.						
Access	Public						
Syntax	<i>instancename.of_Add (newnode)</i>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_list</td></tr> <tr> <td><i>newnode</i></td><td>N_cst_linkedlistnode instance to be added to the linked list</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_list	<i>newnode</i>	N_cst_linkedlistnode instance to be added to the linked list
Argument	Description						
<i>instancename</i>	Instance name of n_cst_list						
<i>newnode</i>	N_cst_linkedlistnode instance to be added to the linked list						
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.						
Usage	You must instantiate and initialize <i>newnode</i> before calling this function.						
Examples	This example calls the of_Add function:						

```
n_cst_linkedlistnode lnv_node
Integer li_return

lnv_node = CREATE n_cst_linkedlistnode

lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_list.of_Add(lnv_node)
```

of_AddHead

Description	Adds a new node to the beginning of the linked list.
Access	Public
Syntax	<i>instancename.of_AddHead (newnode)</i>

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_list</i>
<i>newnode</i>	N_cst_linkedlistnode instance to be added at the beginning of the linked list

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

Usage Internal.

Examples This example calls the of_AddHead function:

```
...
if isnull(inv_head) or not isvalid(inv_head) then
    return of_AddHead(anv_newnode)
end if
...
```

of_AddTail

Description Adds a new node to the end of the linked list.

Access Public

Syntax *instancename.of_AddTail (newnode)*

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_list</i>
<i>newnode</i>	N_cst_linkedlistnode instance to be added at the end of the linked list

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

Usage Internal.

Examples This example calls the of_AddTail function:

```
...
if anv_insertafternode = inv_tail Then
    Return of_AddTail(anv_newnode)
End If
...
```

of_Find

Description Retrieves a node from the linked listed based upon the key value.

Access Public

Syntax

instancename.of_Find (foundnode, nodewithkey)

Argument	Description
<i>instancename</i>	Instance name of n_cst_list
<i>foundnode</i>	N_cst_linkedlistnode instance into which the function places the node if a match is found (passed by reference)
<i>nodewithkey</i>	N_cst_linkedlistnode instance containing the key value used to find the requested node

Return value

Integer. Returns values as follows:

- **1** Success
- **0** Node not found
- **-1** Error

Examples

This example is from the of_IsInList function:

```
...
If of_Find(lnv_locatenode, anv_node) = 1 Then
    // Node was found in list.
    Return True
End If
...
```

of_IsDuplicatesAllowed**Description**

Reports whether a sorted linked list allows duplicates.

Access

Public

Syntax

instancename.of_IsDuplicatesAllowed ()

Argument	Description
<i>instancename</i>	Instance name of n_cst_list

Return value

Boolean. Returns TRUE if duplicates are allowed and FALSE if they are not.

Examples

This example calls the of_IsDuplicatesAllowed function:

```
Boolean lb_duplicates

lb_duplicates = inv_list. of_IsDuplicatesAllowed()
IF lb_duplicates THEN
    MessageBox("Sorted List", "Duplicates are allowed")
ELSE
    MessageBox("Sorted List", "Duplicates not allowed")
END IF
```

of_IsInList

Description Reports whether a node with the specified key is in the list.

Access Public

Syntax *instancename.of_IsInList (node)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_list
<i>node</i>	N_cst_linkedlistnode instance containing the key used as search criteria

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

Usage This function calls the of_Find function.

Examples This example calls the of_IsInList function:

```
n_cst_linkedlistnode  lnv_temp
Boolean  lb_return

lnv_temp = CREATE n_cst_linkedlistnode
lnv_temp.of_SetKey(sle_2.text)
lb_return = inv_list.of_IsInList(lnv_temp)
IF lb_return THEN
    MessageBox( "Found" , "Node was found" )
ELSE
    MessageBox( "Not Found" , "Node was not found" )
END IF
```

of_IsSorted

Description Reports whether a list is sorted.

Access Public

Syntax *instancename.of_IsSorted ()*

Argument	Description
<i>instancename</i>	Instance name of n_cst_list

Return value Boolean. Returns TRUE if the list is sorted and FALSE if it is not.

Examples

This example calls the of_IsSorted function:

```
IF inv_list.of_IsSorted() THEN
    MessageBox("Linked List", "Sorted")
ELSE
    MessageBox("Linked List", "Unsorted")
END IF
```

of_Remove**Description**

Removes the specified node from a linked list, resetting pointers as appropriate.

Access

Public

Syntax

instancename.of_Remove (removenode)

Argument	Description
<i>instancename</i>	Instance name of n_cst_list
<i>removenode</i>	N_cst_linkedlistnode instance to remove from the linked list

Return value

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

Examples

This example calls the of_Remove function:

```
n_cst_linkedlistnode lnv_node, lnv_temp
Integer li_return
Long ll_oldtextcolor

lnv_temp = CREATE n_cst_linkedlistnode

IF sle_2.text = "" THEN
    MessageBox("Data Entry Error", &
               "You must enter a node to delete")
    Return
END IF

lnv_temp.of_SetKey(sle_2.text)
li_return = inv_list.of_Find(lnv_node, lnv_temp)
IF li_return < 1 THEN
    ll_oldtextcolor = sle_2.textcolor
    sle_2.textcolor = RGB(255, 0, 0)
    MessageBox("Not Found", &
               "The specified node was not found")
    sle_2.textcolor = ll_oldtextcolor
    Return
END IF
li_return = inv_list.of_Remove(lnv_node)
```

```
IF li_return = 1 THEN
    MessageBox( "Success" , "Node removed successfully" )
END IF
```

of_RemoveHead

Description

Removes the first node from a linked list, resetting pointers as appropriate and returning the removed node.

Access

Public

Syntax

instancename.of_RemoveHead (removenode)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_list</i>
<i>removenode</i>	<i>N_cst_linkedListnode</i> instance into which the function places a reference to the returned node (passed by reference)

Return value

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

Usage

Internal.

Examples

This example calls the *of_RemoveHead* function:

```
Return Super::of_RemoveHead(anv_headnode)
```

of_RemoveTail

Description

Removes the last node from a linked list, resetting pointers as appropriate and returning the removed node.

Access

Public

Syntax

instancename.of_RemoveTail (removenode)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_list</i>
<i>removenode</i>	<i>N_cst_linkedListnode</i> instance into which the function places a reference to the returned node (passed by reference)

Return value

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

Usage

Internal.

Examples

This example calls the *of_RemoveTail* function:

```
Return Super::of_RemoveTail(anv_tailnode)
```

of_SetDuplicatesAllowed

Description	Specifies whether a sorted list allows duplicates.						
Access	Public						
Syntax	<i>instancename.of_SetDuplicatesAllowed (boolean)</i>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_list</td></tr> <tr> <td><i>boolean</i></td><td>Boolean indicating whether duplicates are allowed (TRUE) or not (FALSE)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_list	<i>boolean</i>	Boolean indicating whether duplicates are allowed (TRUE) or not (FALSE)
Argument	Description						
<i>instancename</i>	Instance name of n_cst_list						
<i>boolean</i>	Boolean indicating whether duplicates are allowed (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_SetDuplicatesAllowed function:

```
...
inv_list.of_SetSorted(TRUE)
inv_list.of_SetDuplicatesAllowed(FALSE)
...
```

of_SetSorted

Description	Specifies whether the list is sorted.				
Access	Public				
Syntax	<i>instancename.of_SetSorted ()</i>				
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_list</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_list
Argument	Description				
<i>instancename</i>	Instance name of n_cst_list				

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

Usage For a sorted list, you should also call of_SetDuplicatesAllowed to specify whether the sorted list allows duplicates.

Examples This example calls the of_SetSorted function:

```
...
inv_list.of_SetSorted(TRUE)
inv_list.of_SetDuplicatesAllowed(FALSE)
...
```

n_cst_luw

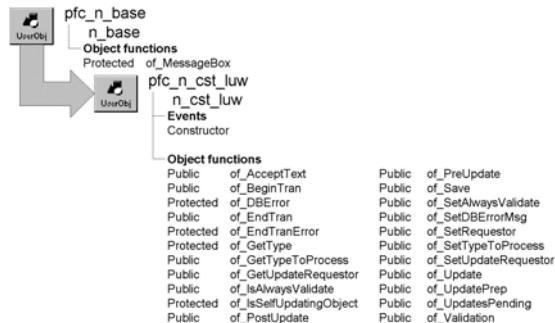
Description

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.

See the *PFC User's Guide*.

See also

n_ds
u_base

```

u_dw
u_lvs
u_tab
u_tvs
w_master

```

Instance variables

`N_cst_luw` includes instance variables:

Instance variable	Description	Data type	Access	Usage
<code>ALL_OBJECTS</code>	Constant set to an empty string	String	Public	This is the default for <code>ib_typetoprocess</code>
<code>ib_accepttext</code>	Indicates whether the <code>of_Save</code> function accepts text by default	Boolean	Protected	The default is TRUE
<code>ib_alwaysvalidate</code>	Indicates whether <code>n_cst_luw</code> validates all controls, even those with no pending updates	Boolean	Protected	Set with <code>of_SetAlwaysValidate</code> (the default is FALSE)
<code>ib_resetflag</code>	Indicates whether the <code>of_Save</code> function resets update flags by default	Boolean	Protected	The default is FALSE
<code>inv_dberrorattrib</code>	Database error information	<code>n_cst_dberrorattrib</code>	Protected	Set with <code>of_SetDBErrorMsg</code>
<code>inv_metaclass</code>	Reference to metaclass service	<code>n_cst_metaclass</code>	Protected	Internal
<code>ipo_pendingupdates[]</code>	Controls with pending updates	PowerObject	Protected	Set by <code>of_UpdatesPending</code>
<code>ipo_requestor</code>	Object owning this instance of <code>n_cst_luw</code>	PowerObject	Protected	Set with <code>of_Requestor</code>
<code>ipo_updaterequestor</code>	Object requesting the update	PowerObject	Protected	Set with <code>of_SetUpdateRequestor</code>
<code>is_instancename</code>	Unique name for this instance of <code>n_cst_luw</code>	String	Protected	Internal
<code>is_typetoprocess</code>	Customized list of object types to update	String	Protected	Set with <code>of_SetTypeToProcess</code> . Default is <code>ALL_OBJECTS</code>

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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>controlarray</i>	PowerObject array containing the control to check
<i>focusonerror</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>transactions</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>errorinfo</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>transactions</i>	N_tr array containing transactions
<i>savestate</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>trcommit</i>	N_tr array referencing transactions that were committed
<i>trrollback</i>	N_tr array referencing transactions that were rolled back
<i>returncodes</i>	Integer array containing ROLLBACK return codes
<i>savestate</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>control</i>	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
<i>instancename</i>	Instance name of <i>n_cst_luw</i> (the PFC default is <i>inv_luw</i>)

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
<i>instancename</i>	Instance name of <i>n_cst_luw</i> (the PFC default is <i>inv_luw</i>)
<i>requestor</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>control</i>	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
<i>instancename</i>	Instance name of <i>n_cst_luw</i> (the PFC default is <i>inv_luw</i>)
<i>controls</i>	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 update 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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>control</i>	PowerObject or PowerObject array containing the control or controls to be updated
<i>transaction</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>errorinfo</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>requestor</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>controltypes</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>requestor</i>	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 do not 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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>controlarray</i>	PowerObject array containing the controls to be updated
<i>accepttext</i>	Boolean indicating whether to perform an accept text before performing the update
<i>resetflags</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the PFC default is inv_luw)
<i>controlarray</i>	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
<i>instancename</i>	Instance name of n_cst_luw (the w_master default is inv_luw)
<i>controlarray</i>	PowerObject array containing the controls to be updated
<i>pendingarray</i>	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	<i>instancename.of_Validation (controlarray)</i>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_luw (the w_master default is inv_luw)</td></tr> <tr> <td><i>controlarray</i></td><td>PowerObject array containing controls to be validated</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_luw (the w_master default is inv_luw)	<i>controlarray</i>	PowerObject array containing controls to be validated
Argument	Description						
<i>instancename</i>	Instance name of n_cst_luw (the w_master default is inv_luw)						
<i>controlarray</i>	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:						

```

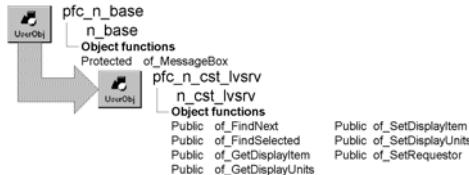
...
If IsValid(inv_luw) Then
    Return inv_luw.of_Validation(apo_control)
End If
...

```

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

N_cst_lvsrv includes instance variables:

Instance variable	Description	Data type	Access	Usage
<i>is_displayitem</i>	Controls message text displayed for a single row	String	Protected	Call of <i>_SetDisplayItem</i> and of <i>_GetDisplayItem</i> to access
<i>is_displayunits</i>	Controls message text displayed for multiple rows	String	Protected	Call of <i>_SetDisplayUnits</i> and of <i>_GetDisplayUnits</i> 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	<code>lvcontrol.instancename.of_FindNext (startindex, direction)</code>										
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the u_lvs-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)</td></tr> <tr> <td><i>startindex</i></td><td>Integer specifying the index number from which you want your search to begin</td></tr> <tr> <td><i>direction</i></td><td>Direction enumerated data type specifying the search direction: DirectionAll! DirectionUp! DirectionDown! DirectionLeft! DirectionRight!</td></tr> </tbody> </table>	Argument	Description	<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control	<i>instancename</i>	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)	<i>startindex</i>	Integer specifying the index number from which you want your search to begin	<i>direction</i>	Direction enumerated data type specifying the search direction: DirectionAll! DirectionUp! DirectionDown! DirectionLeft! DirectionRight!
Argument	Description										
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control										
<i>instancename</i>	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)										
<i>startindex</i>	Integer specifying the index number from which you want your search to begin										
<i>direction</i>	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 <i>startindex</i> . 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

lvcontrolinstancename.of_FindSelected (startindex, direction)

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv</i> (the u_lvs default for this is <i>inv_base</i>)
<i>startindex</i>	Integer specifying the index number from which you want your search to begin
<i>direction</i>	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	<code>lvcontrol.instancename.of_GetDisplayItem()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the u_lvs-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)</td></tr> </tbody> </table>	Argument	Description	<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control	<i>instancename</i>	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
Argument	Description						
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control						
<i>instancename</i>	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 <i>this row</i> , 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	<code>lvcontrol.instancename.of_GetDisplayUnits()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the u_lvs-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)</td></tr> </tbody> </table>	Argument	Description	<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control	<i>instancename</i>	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
Argument	Description						
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control						
<i>instancename</i>	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 <i>rows</i> , 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

lvcontrolinstancename.of_SetDisplayItem (itemname)

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv</i> (the u_lvs default for this is <i>inv_base</i>)
<i>itemname</i>	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

lvcontrolinstancename.of_SetDisplayUnits (units)

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv</i> (the u_lvs default for this is <i>inv_base</i>)
<i>units</i>	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
<i>lvcontrol</i>	Instance name of the <code>u_lvs</code> -based ListView control
<i>instancename</i>	Instance name of <code>n_cst_lvsrv</code> (the <code>u_lvs</code> default for this is <code>inv_base</code>)
<i>requestor</i>	<code>U_lvs</code> -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
...

```

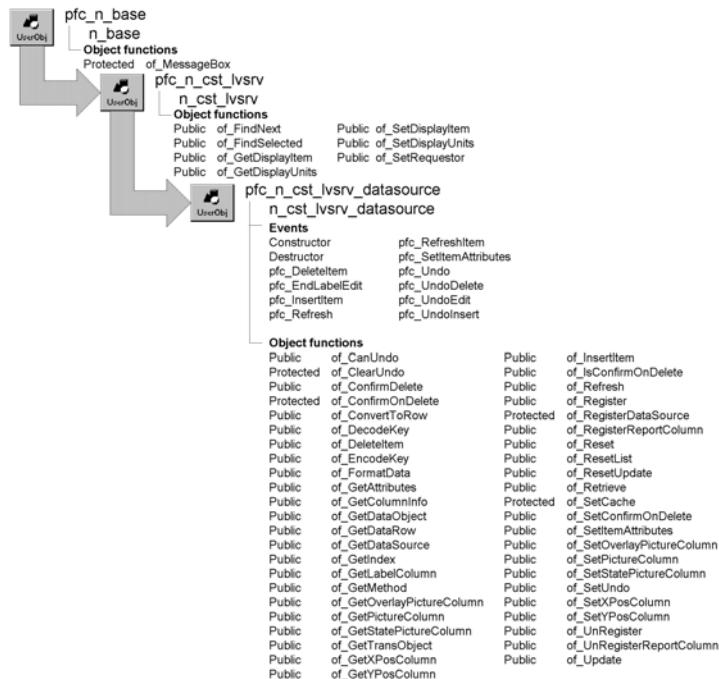
n_cst_lvsrv_datasource

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_cst_columnnattrib`
`n_cst_lvsrvattrib`
`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()
	5 Extend the u_lvs pfc_Retrieve event to retrieve rows: Any la_args[20] n_ds lds_data la_args[1] = "windows" // Retrieval argument this.of_Retrieve(la_args, lds_data)
See also	n_cst_lvsrv_sort n_cst_tvsrv_levelsource u_lvs

Instance variables

N_cst_lvsrv_datasource includes instance variables:

Instance variable	Description	Data type	Access	Usage
CACHE_ID	Constant set to "pfc listview"	String	Public	Internal
ib_confirmdelete	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_isundoing	Indicates whether an undo is occurring	Boolean	Protected	Internal
ib_undo	Indicates whether undo is supported	Boolean	Protected	Set with of_SetUndo

Instance variable	Description	Data type	Access	Usage
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
is_pfckey	Key value	String	Protected	Internal
is_undoeeditlabeltext	Modified label text	String	Protected	Internal
is_undotype	Undo type	String	Protected	Internal
inv_attrib	ListView attributes	n_cst_lvsrvattrib	Protected	Set with of_Register
inv_colattrib[]	Column attributes	n_cst_columnnattrib	Protected	Set with of_RegisterReportColumn
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 pre-coded events:

Constructor	pfc_RefreshItem
Destructor	pfc_SetItemAttributes
pfc_DeleteItem	pfc_Undo
pfc_EndLabelEdit	pfc_UndoDelete
pfc_InsertItem	pfc_UndoEdit
pfc_Refresh	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 Destroyed.

pfc_DeleteItem

Description	Deletes a row from the DataStore.
Syntax	<code>lvcontrol.instancename.EVENT pfc_DeleteItem (index)</code>
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)
<i>index</i>	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 remove the row from the database. To remove the row from the database, call the PowerScript Update function. To get a reference to the DataStore, call <code>of_GetAttributes</code> and use the <code>ids_source</code> instance variable in the returned <code>n_cst_lvsrvattrib</code> instance. 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	<code>lvcontrol.instancename.EVENT pfc_EndLabelEdit (index, newlabel)</code>
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this value is inv_datasource)
<i>index</i>	Integer specifying the index of the modified ListView item
<i>newlabel</i>	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_InsertItem

Description Inserts a new item into the ListView.

Syntax *lvcontrol.instancecname.EVENT pfc_InsertItem (datastore, row, lvitem, position, index)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancecname</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this value is <i>inv_datasource</i>)
<i>datastore</i>	N_ds containing the data for the new item. This can be either the DataStore maintained by <i>n_cst_lvsrv_datasource</i> 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 <i>n_cst_lvsrv_datasource</i> . This argument is accessed through the <i>ads_obj</i> argument
<i>row</i>	Long specifying the row containing the item to be inserted. This argument is accessed through the <i>al_row</i> argument
<i>lvitem</i>	ListViewItem containing the item to be inserted. This argument is accessed through the <i>alvi_item</i> argument
<i>position</i>	String specifying the position in which to insert the new item Values are: <ul style="list-style-type: none">• First• Last• Sort• After If you specify After, you must also specify <i>index</i> This argument is accessed through the <i>as_position</i> argument

	Argument	Description
	<i>index</i>	Integer containing the index to the ListView item before the new item. The function uses this argument if you specify After for <i>position</i> . This argument is accessed through the <i>ai_item</i> argument
Return value		Integer. Returns the index of the new item if the function succeeds and -1 if an error occurs.
Usage		This event calls the <code>of_InsertItem</code> function.
Examples		This example is from the <code>u_lvs pfc_InsertItem</code> event:
		...
		<code>Return inv_datasource.event pfc_InsertItem &</code>
		<code>(ads_source, al_row, llvi_new, as_position, ai_index)</code>
		...

pfc_Refresh

Description	Refreshes the ListView items with data from the data source.
Return value	Integer. Returns the number of items refreshed if the event succeeds and -1 if an error occurs.
Examples	This example is from the <code>u_lvs Refresh</code> event:
<pre>If isvalid(inv_datasource) then Return inv_datasource.event pfc_Refresh() End If ...</pre>	

pfc_RefreshItem

Description	Refreshes the specified ListView item, resetting all properties to the defaults specified in the data source.								
Syntax	<code>lvcontrol.instancename.EVENT pfc_RefreshItem (<i>index</i>)</code>								
	Argument								
<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the <code>u_lvs</code>-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of <code>n_cst_lvsrv_datasource</code> (the <code>u_lvs</code> default for this value is <code>inv_datasource</code>)</td></tr> <tr> <td><i>index</i></td><td>Integer specifying the ListView index of the item to refresh</td></tr> </tbody> </table>		Argument	Description	<i>lvcontrol</i>	Instance name of the <code>u_lvs</code> -based ListView control	<i>instancename</i>	Instance name of <code>n_cst_lvsrv_datasource</code> (the <code>u_lvs</code> default for this value is <code>inv_datasource</code>)	<i>index</i>	Integer specifying the ListView index of the item to refresh
Argument	Description								
<i>lvcontrol</i>	Instance name of the <code>u_lvs</code> -based ListView control								
<i>instancename</i>	Instance name of <code>n_cst_lvsrv_datasource</code> (the <code>u_lvs</code> default for this value is <code>inv_datasource</code>)								
<i>index</i>	Integer specifying the ListView index of the item to refresh								
Return value	Integer. Returns 1 if the event succeeds and -1 if an error occurs.								

Examples

This example is from the u_lvs pfc_RefreshItem event:

```
If isvalid(inv_datasource) then
    Return inv_datasource.event pfc_RefreshItem &
        (ai_index)
End If
...
```

pfc_SetItemAttributes

Description

Sets default properties for the ListView item before insertion.

Syntax

lvcontrol.*instancename*.EVENT **pfc_SetItemAttributes** (*data*, *row*, *lvitem*)

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this value is <i>inv_datasource</i>)
<i>data</i>	N_ds containing the data for the item. This argument is accessed through the <i>ads_obj</i> argument
<i>row</i>	Long specifying the row containing the item. This argument is accessed through the <i>al_row</i> argument
<i>lvitem</i>	ListViewItem to be inserted. This argument is accessed through the <i>alvi_item</i> argument (passed by reference)

Usage

This event calls the of_SetItemAttributes function.

pfc_Undo

Description

Restores the last ListView change.

Syntax

lvcontrol.*instancename*.EVENT **pfc_Undo** ()

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this value is <i>inv_datasource</i>)

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 isinvalid(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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	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 *lvcontrolinstancename.EVENT pfc_UndoInsert()*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (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_InsertItem
of_ClearUndo	of_IsConfirmonDelete
of_ConfirmDelete	of_Refresh
of_ConfirmonDelete	of_Register
of_ConvertToRow	of_RegisterDataSource
of_DecodeKey	of_RegisterReportColumn
of_DeleteItem	of_Reset
of_EncodeKey	of_ResetList
of_FormatData	of_ResetUpdate
of_GetAttributes	of_Retrieve
of_GetColumnInfo	of_SetCache
of_GetDataObject	of_SetConfirmonDelete
of_GetDataRow	of_SetItemAttributes
of_GetDataSource	of_SetOverlayPictureColumn
of_GetIndex	of_SetPictureColumn
of_GetLabelColumn	of_SetStatePictureColumn
of_GetMethod	of_SetUndo
of_GetOverlayPictureColumn	of_SetXPosColumn
of_GetPictureColumn	of_SetYPosColumn
of_GetStatePictureColumn	of_UnRegister
of_GetTransObject	of_UnRegisterReportColumn
ofGetXPosColumn	of_Update
ofGetYPosColumn	

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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Argument	Description
<i>undotype</i> (optional)	<p>String into which the function places the type of the last operation (passed by reference). Values are:</p> <ul style="list-style-type: none"> • 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 *lvcontrolinstancename.of_ClearUndo()*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)

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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>amount</i>	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_ConfirmOnDelete**Description**

Reports whether a confirmation dialog box displays when the user deletes a row.

Access

Protected

Syntax`lvcontrol.instancename.of_ConfirmOnDelete ()`

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)

Return value

Boolean. Returns TRUE if confirm on delete is enabled and FALSE if it is disabled.

Usage

Internal.

Examples

This example is from the n_cst_lvsrv_datasource_of_DeleteItem function:

```
...
    if of_ConfirmOnDelete() then
        If li_totalitems = 1 then
            ...

```

of_ConvertToRow

Description	Converts an array of data into a row in the DataStore.
Access	Public
Syntax	<i>lvcontrol.instancename.of_ConvertToRow (colvalues, datastore, row)</i>
<hr/>	
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>colvalues</i>	Any array containing the data to be used for the new row
<i>datastore</i>	N_ds instance into which the function places the new row (passed by reference)
<i>row</i>	Long into which the function places the new row number (passed by reference)
<hr/>	
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_DecodeKey

Description	Retrieves the decoded key from the ListView item's data attribute.
Access	Public
Syntax	<i>lvcontrol.instancename.of_DecodeKey (data)</i>
<hr/>	
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>data</i>	String containing the ListView item's data attribute
<hr/>	
Return value	String. Returns the unique key if the function succeeds and ! if an error occurs.
Usage	Internal.

Examples

This example is from the pfc_UndoDelete event:

```
...
ls_undokey = of_DecodeKey(string(ltvi_undo.data))
...
```

of_DeleteItem**Description**

Removes either the specified item or all selected items from a ListView and the associated DataStore.

Access

Public

Syntax

lvcontrol.*instancename*.**of_DeleteItem** ({ *index* })

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>index</i> (optional)	Integer specifying the index of the item to be deleted

Return value

Integer. Returns 1 if the function succeeds, 0 if nothing was deleted, and -1 if an error occurs.

Usage

The pfc_DeleteItem event calls this function.

Examples

This example is from the pfc_DeleteItem event:

```
...
return of_DeleteItem()
...
```

of_EncodeKey**Description**

Creates an encoded key from the ListView item's data attribute. The service uses this key to access rows in the associated DataStore.

Access

Public

Syntax

lvcontrol.*instancename*.**of_EncodeKey** (*source*, *row*)

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>source</i>	N_ds containing the DataStore to populate the item
<i>row</i>	Long specifying the row containing the item

Return value	String. Returns the unique key if the function succeeds and ! if an error occurs.
Usage	Internal.
Examples	This example is from the of_InsertItem function:

```
...
ls_key = of_EncodeKey(inv_attrib.ids_source, al_row)
...
```

of_FormatData

Description	Converts DataWindow data from its native format to the string data type.
Access	Public
Syntax	<i>lvcontrol.instancename.of_FormatData (columnname, format, type, datastore, row)</i>

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
<i>columnname</i>	String specifying the column
<i>format</i>	String specifying the column's format
<i>type</i>	String specifying the column's data type
<i>datastore</i>	N_ds specifying the DataStore containing the data
<i>row</i>	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_GetAttributes

Description Retrieves the properties used to populate the ListView's data source.

Access Public

Syntax ***lvcontrol.instancename.of_GetAttributes (/vattrib)***

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>vattrib</i>	N_cst_lvsrvattrib into which the function places ListView properties (passed by reference)

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

Usage Call this function to retrieve a reference to the DataStore and Transaction object for the data source.

Examples This example calls the of_GetAttributes function:

```
n_cst_lvsrvattrib lnv_attrib

lv_1.Event pfc_DeleteItem()
lv_1.inv_datasource.of_Update(TRUE, TRUE)
lv_1.inv_datasource.of_GetAttributes &
(lnv_attrib)
IF Not IsNull(lnv_attrib.itr_obj) THEN
    lnv_attrib.itr_obj.of_Commit()
END IF
...
```

of_GetColumnInfo

Description Retrieves column information from the ListView.

Access Public

Syntax ***lvcontrol.instancename.of_GetColumnInfo (column, colattrib)***

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv (the u_lvs default for this is inv_base)
<i>column</i>	Integer specifying the column from the ListView
<i>colattrib</i>	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	<i>lvcontrolinstancename.of.GetDataObject ()</i>

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	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	<i>lvcontrolinstancename.of.GetDataRow (item, datasource, row)</i>

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>item</i>	Integer specifying the index to the ListView item

Argument	Description
<i>datasource</i>	N_ds specifying the DataStore that contains the row (passed by reference)
<i>row</i>	Long specifying the row in <i>datasource</i> that corresponds to <i>item</i> (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(li_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 *lvcontrolinstancename.of_GetDataSource (source)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>source</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>row</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)

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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	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

lvcontrolinstancename.of_GetPictureColumn ()

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (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

lvcontrolinstancename.of_GetStatePictureColumn ()

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>transaction</i>	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
```

ofGetXPosColumn**Description**

Retrieves the name of the column that specifies the item's x coordinate.

Access

Public

n_cst_lvsrv_datasource

Syntax	<code>lvcontrol.instancecname.of_GetXPosColumn ()</code>
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancecname</i>	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: <pre>String ls_column ls_column = lv_1.inv_datasource.of_GetXPosColumn() MessageBox("ListView", "XPos column is: " & + ls_column)</pre>

of_GetYPosColumn

Description	Retrieves the name of the column that specifies the item's y coordinate.
Access	Public
Syntax	<code>lvcontrol.instancecname.of_GetYPosColumn ()</code>
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancecname</i>	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: <pre>String ls_column ls_column = lv_1.inv_datasource.of_GetYPosColumn() MessageBox("ListView", "YPos column is: " & + ls_column)</pre>

of_InsertItem

Description	Adds a new item to the ListView using a row from the data source.
Access	Public

Syntax *lvcontrolinstancename.of_InsertItem (datastore, row, lvitem, position, item)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>datastore</i>	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
<i>row</i>	Long specifying the row containing the new item
<i>lvitem</i>	ListViewItem containing the item to be added to the ListView
<i>position</i>	String specifying the position under the current parent in which to insert the new item. Values are: <ul style="list-style-type: none"> • First • Last • Before • After
<i>item</i>	Integer containing an index relative to the current position. The function uses this argument if you specify Before or After for <i>position</i>

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

n_cst_lvsrv_datasource

Syntax	<code>lvcontrol.instance.name.of_IsConfirmonDelete()</code>
Argument	Description
<code>lvcontrol</code>	Instance name of the u_lvs-based ListView control
<code>instance.name</code>	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:
	<pre>IF lv_1.inv_datasource.of_IsConfirmonDelete() THEN MessageBox("ListView", "Confirm enabled") ELSE MessageBox("ListView", "Confirm disabled") END IF</pre>

of_Refresh

Description	Updates the ListView with items from the data source.
Access	Public
Syntax	<code>lvcontrol.instance.name.of_Refresh()</code>
Argument	Description
<code>lvcontrol</code>	Instance name of the u_lvs-based ListView control
<code>instance.name</code>	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

To	Use
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

Description

To display data retrieved via a DataWindow object

Establishes a ListView data source by retrieving rows using the specified DataWindow object.

Access

Public

Syntax

```
lvcontrolinstancename.of_Register ( labelcolumn, dwobject
{, trans } )
```

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>labelcolumn</i>	String specifying the <i>dwobject</i> column that u_lvs uses for ListView item labels
<i>dwobject</i>	String specifying the DataWindow object used to retrieve data
<i>trans</i> (optional)	N_tr Transaction object used to retrieve rows for <i>dwobject</i> If you do not specify this argument, <i>dwobject</i> 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

Description Establishes a ListView data source by retrieving rows from the database using the specified SQL statement.

Access Public

Syntax *lvcontrolinstancename.of_Register (labelcolumn, trans, sql)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>labelcolumn</i>	String specifying the column that u_lvs uses for ListView item labels
<i>trans</i>	N_tr Transaction object used to retrieve rows for <i>sql</i>
<i>sql</i>	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
<code>lvcontrol</code>	Instance name of the u_lvs-based ListView control
<code>instancename</code>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is <code>inv_datasource</code>)
<code>labelcolumn</code>	String specifying the column that u_lvs uses for ListView item labels
<code>dwcontrol</code>	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

lvcontrolinstancename.of_Register (labelcolumn, datastore)

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>labelcolumn</i>	String specifying the column that <i>u_lvs</i> uses for ListView item labels
<i>datastore</i>	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	<code>lvcontrol.instancename.of_Register (labelcolumn, dwobject, data)</code>
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>labelcolumn</i>	String specifying the column that u_lvs uses for ListView item labels
<i>dwobject</i>	String specifying the DataWindow object whose columns correspond to <i>data</i>
<i>data</i>	PowerObject array containing the data to display in the ListView
Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none"> • 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.
<hr/>	
Elements must match	
Errors will occur if the elements of the <i>data</i> structure do not exactly match the columns of <i>dwobject</i> .	
<hr/>	
The of_Refresh function does not affect data cached with this syntax.	
Examples	This example calls the of_Register function:
<pre> 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] </pre>	

```
        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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>labelcolumn</i>	String specifying the column that u_lvs uses for ListView item labels
<i>dwoBJECT</i>	String specifying the DataWindow object whose columns correspond to the data in <i>file</i>
<i>file</i>	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 *lvcontrolinstancename.of_RegisterDataSource (method, dwobject, trans, sql, data, dwcontrol, datastore, file)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>method</i>	String specifying the data source population method. Values are: Retrieve DataObjectData PowerObject DataWindowControl DataStoreControl ImportFile SQL
<i>dwobject</i>	String specifying the DataWindow object used by the data source
<i>trans</i>	N_tr specifying the Transaction object used by the data source
<i>sql</i>	String specifying the SQL statement used as the data source
<i>data</i>	PowerObject array containing data used as the data source
<i>dwcontrol</i>	DataWindow control whose associated DataWindow object and data is used as the data source
<i>datastore</i>	DataStore whose associated DataWindow object and data is used as the data source

Argument	Description
<i>file</i>	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 *lvcontrolinstancename.of_RegisterReportColumn ()*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (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	<code>lvcontrol.instancename.of_Reset()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the u_lvs-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)</td></tr> </tbody> </table>	Argument	Description	<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control	<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
Argument	Description						
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control						
<i>instancename</i>	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	<code>lvcontrol.instancename.of_ResetList()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the u_lvs-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)</td></tr> </tbody> </table>	Argument	Description	<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control	<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
Argument	Description						
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control						
<i>instancename</i>	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

n_cst_lvsrv_datasource

Syntax	<code>lvcontrol.instance.name.of_ResetUpdate()</code>
Argument	Description
<code>lvcontrol</code>	Instance name of the u_lvs-based ListView control
<code>instance.name</code>	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.instance.name.of_Retrieve(arguments, data)`

Argument	Description
<code>lvcontrol</code>	Instance name of the u_lvs-based ListView control
<code>instance.name</code>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<code>arguments</code>	Twenty-element Any array containing retrieval arguments
<code>data</code>	N_ds into which the function retrieves data (passed by reference)

Return value Long. 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
    Return = inv_datasource.of_Retrieve &
    (aa_args, ads_data)
End If
...
```

of_SetCache

Description Enables or disables n_cst_dwcache.

Access Protected

Syntax

```
lvcontrol.instancename.of_SetCache ( boolean )
```

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>boolean</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>boolean</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>datastore</i>	N_ds containing the DataStore used to create ListView item properties
<i>row</i>	Long specifying the row in <i>datastore</i> used to create ListView item properties
<i>item</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>column</i>	String specifying one of the following: <ul style="list-style-type: none">• 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 views.

Access Public

Syntax *lvcontrolinstancename.of_SetPictureColumn (column)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>column</i>	String specifying one of the following: <ul style="list-style-type: none"> • 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	<code>lvcontrol.instancename.of_SetStatePictureColumn (column)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>lvcontrol</i></td><td>Instance name of the u_lvs-based ListView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)</td></tr> <tr> <td><i>column</i></td><td> <p>String specifying one of the following:</p> <ul style="list-style-type: none"> 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 </td></tr> </tbody> </table>	Argument	Description	<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control	<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)	<i>column</i>	<p>String specifying one of the following:</p> <ul style="list-style-type: none"> 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
Argument	Description								
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control								
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)								
<i>column</i>	<p>String specifying one of the following:</p> <ul style="list-style-type: none"> 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_SetUndo

Description Specifies whether n_cst_lvsrv_datasource enables undo processing.

Access Public

Syntax

`lvcontrol.instancename.of_SetUndo (boolean)`

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>boolean</i>	Boolean specifying whether to enable (TRUE) or disable (FALSE) undo capabilities

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

Examples

This example calls the of_SetUndo function:

```

...
this.inv_datasource.of_SetUndo(FALSE)
...
```

of_SetXPosColumn

Description	Specifies the DataWindow object column that u_lvs uses for the item's x coordinate.
Access	Public
Syntax	<code>lvcontrol.instancename.of_SetXPosColumn (column)</code>
<hr/>	
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>column</i>	String specifying the DataWindow object column that specifies the item's x coordinate
<hr/>	
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	<code>lvcontrol.instancename.of_SetYPosColumn (column)</code>
<hr/>	
Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>column</i>	String specifying the DataWindow object column that specifies the item's y coordinate
<hr/>	
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_UnRegister

Description Removes a data source.

Access Public

Syntax ***lvcontrol.instancecname.of_UnRegister ()***

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancecname</i>	Instance name of <i>n_cst_lvsrv_datasource</i> (the u_lvs default is <i>inv_datasource</i>)

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

Examples This example calls the of_UnRegister function:

```
...  
lv_1.inv_datasource.of_UnRegister()  
...
```

of_UnRegisterReportColumn

Description Removes the specified column or all columns from the ListView.

Access Public

Syntax ***lvcontrol.instancecname.of_UnRegisterReportColumn ({ column })***

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancecname</i>	Instance name of <i>n_cst_lvsrv</i> (the u_lvs default for this is <i>inv_datasource</i>)
<i>column</i> (optional)	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.inv_datasource.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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_datasource (the u_lvs default for this is inv_datasource)
<i>accept</i>	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database
<i>resetflag</i>	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.

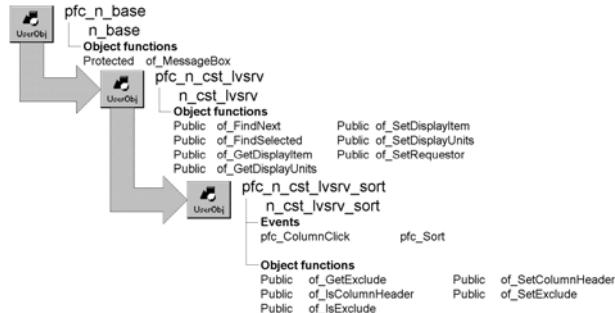
Examples This example calls the of_Update function:

```
lv_1.inv_datasource.of_Update(TRUE, TRUE)
```

n_cst_lsrvr_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_lsrvr_datasource](#)
[u_lvs](#)

Instance variables

N_cst_lsrvr_sort includes 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_excludedcolumns[]	Columns for which sorting is disabled	String	Protected	Set with of_SetExclude

Events

N_cst_lsrvr_sort includes pre-coded 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 *lvcontrolinstancename.EVENT pfc_ColumnClick (column)*

Argument	Description
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lsrvr_sort (the u_lvs default for this value is inv_sort)
<i>column</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lsrv_sort</i> (the u_lvs default for this value is <i>inv_sort</i>)
<i>index1</i>	Integer specifying the index of the first item being compared
<i>index2</i>	Integer specifying the index of the second item being compared
<i>column</i>	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_lsrv_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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
<i>columns</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_sort</i> (the u_lvs default for this value is <i>inv_sort</i>)
<i>column</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of <i>n_cst_lvsrv_sort</i> (the u_lvs default for this value is <i>inv_sort</i>)
<i>boolean</i>	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
<i>lvcontrol</i>	Instance name of the u_lvs-based ListView control
<i>instancename</i>	Instance name of n_cst_lvsrv_sort (the u_lvs default for this value is inv_sort)
<i>columns</i>	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_IsInArray	of_TriggerEvent
-------------------------------------	-----------------

of_GetAllToolbarIndex

Description Retrieves an array containing all unique toolbar item index values.

Access Public

Syntax *instancename.of_GetAllToolbarIndex (menu, index)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_menu
<i>menu</i>	Menu variable containing the menu whose toolbar index is returned
<i>index</i>	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
<i>instancename</i>	Instance name of n_cst_menu
<i>menu</i>	Menu variable containing the menu for which to find the MDI frame
<i>frame</i>	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
<i>instancename</i>	Instance name of <i>n_cst_menu</i>
<i>menu</i>	Menu variable containing the menu for which to retrieve the menu item reference
<i>itemname</i>	String specifying the menu item name
<i>itemref</i>	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
<i>instancename</i>	Instance name of n_cst_menu
<i>array</i>	Integer array containing currently stored ToolbarItemBarIndex numbers
<i>index</i>	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
<i>instancename</i>	Instance name of n_cst_menu
<i>message</i>	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
<i>instancename</i>	Instance name of <code>n_cst_menu</code>
<i>menu</i>	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
<i>instancename</i>	Instance name of <code>n_cst_menu</code>
<i>menu</i>	Menu variable containing the menu for which to trigger the event
<i>event</i>	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:

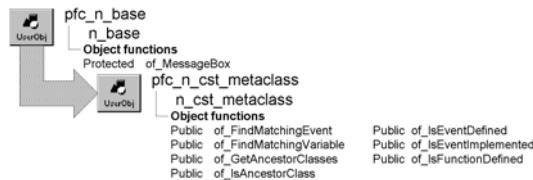
```
n_cst_menu  lnv_menu
Integer      li_return

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
<i>instancename</i>	Instance name of n_cst_metaclass
<i>object</i>	ClassDefinition specifying the object containing <i>event</i>
<i>event</i>	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
<i>instancename</i>	Instance name of n_cst_metaclass
<i>classorscript</i>	ClassDefinition of the object containing <i>variable</i> or ScriptDefinition of the script containing <i>variable</i>
<i>variable</i>	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 lscrdd_object
n_cst_metaclass lnv_metaclass

lcd_object = findClassDefinition("pfc_u_dw")
ls_argtype[1] = "Boolean"
ls_argtype[2] = "Boolean"
lscrdd_object = lcd_object.findMatchingFunction  &
("of_Update", ls_argtype[ ])
if isNull (lscrdd_object) or not isValid  &
(lscrdd_object) then
    messagebox ("", "can't find script definition")
end if
lvrd_object = lnv_metaclass.of_findMatchingVariable &
(lscrdd_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	<i>instancename.of_GetAncestorClasses (sourceclass, ancestorclass)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_metaclass
<i>sourceclass</i>	String or ClassDefinition specifying the class for which to retrieve ancestor classes
<i>ancestorclass</i>	ClassDefinition array into which the function places references to ancestor classes (passed by reference)
Return value	Integer. Returns the number of elements in the <i>ancestorclass</i> array if the function succeeds and -1 if an error occurs.
Usage	The elements in <i>ancestorclass</i> 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	<i>instancename.of_IsAncestorClass (sourceclass, ancestorclass)</i>

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_metaclass</i>
<i>sourceclass</i>	String or ClassDefinition containing the possible descendant class
<i>ancestorclass</i>	String containing the class that might be an ancestor of <i>sourceclass</i>

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
<i>instancename</i>	Instance name of <i>n_cst_metaclass</i>
<i>class</i>	String or ClassDefinition to be tested
<i>event</i>	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
<i>instancename</i>	Instance name of n_cst_metaclass
<i>class</i>	String or ClassDefinition to be tested
<i>event</i>	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
<i>instancename</i>	Instance name of n_cst_metaclass
<i>class</i>	String or ClassDefinition to be tested
<i>function</i>	String specifying the function

Return value Boolean. Returns TRUE if *class* contains *function* and FALSE if it does not.

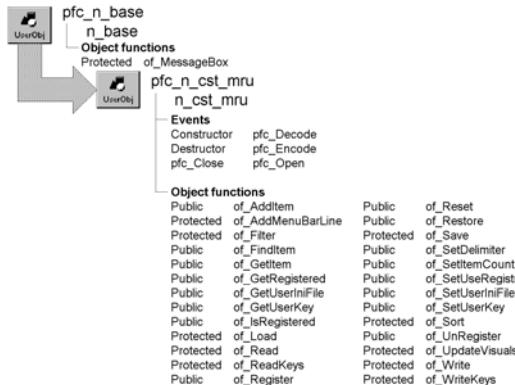
Examples This example is from the n_cst_luw of_AcceptText function:

```
...
lb_defined = inv_metaclass.of_isFunctionDefined &
    (lpo_tocheck.ClassDefinition,  &
     "of_AcceptText", ls_args)
...
```

n_cst_mru

Description	Most recently used (MRU) object service. When enabled, n_cst_mru displays a list of most recently used windows or sheet-specific items on the File menu. By default this list displays up to five items, but you can increase this number. N_cst_mru automatically loads MRU information when the application opens. This information is stored in either the registry (available on Windows platforms) or an INI file (available on all platforms).
-------------	--

Ancestry



Library	PFCAPSRV.PBL PFEAPSRV.PBL
---------	------------------------------

Object relationships	d_mrusevice m_master n_cst_mruattrib n_ds n_cst_menu n_cst_string
----------------------	--

Usage	<p>Use this object to add MRU capabilities to your application's File menu.</p> <p>To use n_cst_mru to open most recently used sheets in an MDI application:</p> <ol style="list-style-type: none"> 1 Enable n_cst_mru by calling the n_cst_appmanager of_SetMRU function. This example is from an application manager Constructor event:
-------	--

```
this.of_SetMRU(TRUE)
```

- 2 Specify where MRU information is to be saved by calling either the n_cst_mru of_SetUserKey (Windows only) or of_SetUserINIFile (all platforms) function. This example, which assumes you've already established the n_cst_appmanager user key or user INI file, saves MRU information in the registry or INI file. This example is from an application manager Constructor event:

```
IF this.of_IsRegistryAvailable( ) THEN
    this.inv_mru.of_SetUserKey &
        (this.of_GetUserKey())
ELSE
    this.inv_mru.of_SetUserINIFile &
        (this.of_GetUserINIFile())
END IF
```

- 3 Register IDs to be tracked the MRU service by calling the n_cst_mru of_Register function. (An ID is the identifier that the window uses to retrieve information through the MRU service.) This is an example of code you can add to the pfc_PreOpen event of the MDI frame window:

```
IF IsValid(gnv_app.inv_mru) THEN
    gnv_app.inv_mru.of_Register( "myapp" )
END IF
```

- 4 Extend the pfc_MRUProcess event in each window that uses the MRU service, adding code to open the appropriate window or sheet (be sure to add this code to the frame window also):

```
Window lw_frame, lw_window
n_cst_menu lnv_menu
n_cst_mruattrib lnv_mruattrib

// Check parameters.
IF IsNull(ai_row) THEN
    Return -1
END IF
IF NOT IsValid(gnv_app.inv_mru) THEN
    Return -1
END IF
// Retrieve row from DataStore.
gnv_app.inv_mru.of_GetItem &
    (ai_row, lnv_mruattrib)
```

```

// Get the MDI frame, if necessary.
lnv_menu.of_GetMDIFrame(this.menuid, lw_frame)

OpenSheet(lw_window, &
          lnv_mruattrib.is_classname, lw_frame)
Return 1

```

Performing other actions in the pfc_MRUProcess event

To see other types of processing you can perform in the pfc_MRUProcess event, see the comments in the ancestor pfc_MRUProcess event.

- 5 Extend the pfc_PreMRUSave event in each window that uses the MRU service. In this event, populate the n_cst_mruattrib object with the ID, classname, key, menu item text, and menu item help to be saved:

```

anv_mruattrib.is_id = "myapp"
anv_mruattrib.is_classname = this.ClassName()
anv_mruattrib.is_menuitemname = this.Title
anv_mruattrib.is_menuitemkey = this.ClassName()
anv_mruattrib.is_menuitemmhelp = &
    "Opens " + this.Title

```

```
Return 1
```

- 6 Extend the pfc_MRURestore event in each window that uses the MRU service. In this event, call the of_Restore function, specifying the ID of the MRU information to display on this window's menu:

```

If IsValid(gnv_app.inv_mru) Then
    Return gnv_app.inv_mru.of_Restore("myapp", This)
End If

```

- 7 Call the pfc_MRUSave event to save MRU information. You can call this event when the window opens, when information is saved, or when the window closes (this example is from the pfc_PreOpen event):

```
this.Event pfc_MRUSave()
```

See also

m_master
n_cst_appmanager
w_master

Instance variables

N_cst_mru includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_useregistry	Reports whether MRU information is stored in the registry or in an INI file	Boolean	Protected	Set with of_SetUseRegistry (default is FALSE)
ids_mrukeys	MRU information	n_ds	Protected	Internal
ii_currentmruccount	Number of items currently in the list	Integer	Protected	Internal
ii_mruiitemcount	Maximum number of MRU items	Integer	Protected	Set with of_SetItemCount (default is 5)
is_delimiter	Separates information stored in the registry or INI file	String	Protected	Set with of_SetDelimiter (default is the semicolon)
is_menubarline	Menu item name for the menu bar line above the MRU list in the File menu	String	Protected	Internal (default is m_pfcmrudash1)
is_menuitemstring	Menu item prefix for items in the MRU list	String	Protected	Internal (default is m_pfcmru)
is_mruccountkey	MRU key name for registry or INI file	String	Protected	Internal (default is ItemCount)
is_mruregisteredsection	MRU section name for registry or INI file	String	Protected	Internal (default is MRU Registered Id)
is_mruiitemkey	MRU key name for registry or INI file	String	Protected	Internal (default is Item)
inv_register[]	Currently currently registered objects	n_cst_mruattrib	Protected	List of registered objects. Add items with of_Register
is_mrusection	MRU section name for registry or INI file	String	Protected	Internal (default is MRU)
is_userinifile	INI file containing registry items	String	Protected	This is typically the user INI file value from n_cst_appmanager
is_userkey	Registry key containing MRU items	String	Protected	This is typically the user key value from n_cst_appmanager

Events

N_cst_mru includes precoded events:

Constructor	pfc_Decode
Destructor	pfc_Encode
pfc_Close	pfc_Open

Constructor

- Description Creates the ids_mrukeys DataStore and assigns the d_mrusevice DataWindow object as its DataObject.
- Usage This event executes when the object is created.

Destructor

- Description Destroys the ids_mrukeys DataStore.
- Usage This event executes when the object is destroyed.

pfc_Close

- Description Saves MRU settings when the application closes.
- Syntax *applicationinstancename.EVENT pfc_Close ()*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_apppreference (the n_cst_appmanager default is inv_mru)

- Usage This event executes when the application closes; it is called by the n_cst_appmanager pfc_Close event if the MRU service is enabled.

pfc_Decode

- Description Breaks down a key value into its component parts.
- Syntax *applicationinstancename.EVENT pfc_Decode (itemkey, mruinfo)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)

Argument	Description
<i>itemkey</i>	String specifying the registry or INI file key to access
<i>mruinfo</i>	N_cst_mruattrib instance into which the event places the class name, menu item text, and key (passed by reference)

- Return value Integer. Returns 1 if the event succeeds and -1 if an error occurs.
- Usage The of_Read function calls this event to access the class name, menu item name, and menu item key for a saved item.
You can extend or override this event to include customized logic accessing the registry or INI file.
To control whether the service accesses information in the registry or in an INI file, call the of_SetUseRegistry function.

pfc_Encode

- Description Combines MRU information into a string to be stored in the registry or INI file.
- Syntax application.instanceclassname.EVENT **pfc_Encode** (*rowindex*, *itemkey*)
- | Argument | Description |
|--------------------------|---|
| <i>application</i> | Global variable of type n_cst_appmanager (must be gnv_app) |
| <i>instanceclassname</i> | Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru) |
| <i>rowindex</i> | Integer specifying the row in ids_mrukeys from which the event accesses information to be saved |
| <i>itemkey</i> | String specifying a key value for the information to be saved |
- Return value Integer. Returns 1 if the event succeeds and -1 if an error occurs.
- Usage The of_Write function calls this event.
You can extend or override this event to include customized logic for storing MRU information in the registry or INI file.

pfc_Open

- Description Initializes object settings and loads the ids_mrukeys DataStore with information from the registry or INI file.

Syntax

applicationinstancename.EVENT pfc_Open (cmdline)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>cmdline</i>	String containing the command line argument passed to the application Open event

Usage

This event executes when the application opens. It is called by the n_cst_appmanager pfc_Open event if the MRU service has been enabled.

Functions

N_cst_mru contains precoded object functions:

of_AddItem	of_Reset
of_AddMenubarLine	of_Restore
of_Filter	of_Save
of_FindItem	of_SetDelimiter
of_GetItem	of_SetItemCount
of_GetRegistered	of_SetUserRegistry
of.GetUserIniFile	of_SetUserIniFile
of.GetUserKey	of_SetUserKey
of_IsRegistered	of_Sort
of_Load	of_UnRegister
of_Read	of_UpdateVisuals
of_ReadKeys	of_Write
of_Register	of_WriteKeys

of_AddItem

Description

Saves information in the ids_mrukeys DataStore.

Access

Public

Syntax	<i>application.instancename.of_AddItem (mruinfo)</i>
Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>mruinfo</i>	N_cst_mruattrib instance containing information to be saved in the ids_mrukeys DataStore
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Usage	The w_master pfc_MRUSave event calls this function to save an MRU item in the DataStore.
Examples	This example is from the w_master pfc_MRUSave event:
	<pre style="font-family: monospace; padding-left: 40px;"> ... if gnv_app.inv_mru.of_IsRegistered & (lnv_mruattrib.is_id) then Return gnv_app.inv_mru.of_additem(lnv_mruattrib) ELSE Return -1 END IF ...</pre>

of_AddMenubarLine

Description Enables display of the separator line below the Exit menu item.

Access Protected

Syntax *application.instancename.of_AddMenubarLine (window)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>window</i>	Window variable whose associated menu is modified

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:

```

...
of_AddMenubarLine(aw_window)
...
```

of_Filter

Description Filters rows in the ids_mrukeys DataStore.

Access Protected

Syntax *application.instancename.of_Filter (filter)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>filter</i>	String specifying the filter. To reset the filter, specify an empty 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_Save function:

```
...
for li_index = 1 to ll_objectcount
    ls_id = inv_register[li_index].is_id
    of_Filter("s_id = '" + ls_id + "'")
    of_Sort("n_identity D")
...

```

of_FindItem

Description Finds menu item text within the ids_mrukeys DataStore.

Access Public

Syntax *application.instancename.of_FindItem (menuitemtext, startrow)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>menuitemtext</i>	String specifying the text to find
<i>startrow</i>	Long specifying the row from which to start searching

Return value Long. Returns the row number if *menuitemtext* is found and -1 if *menuitemtext* is not found.

Usage Internal.

Examples

This example is from the of_AddItem function:

```
...
    li_currentrow = of_finditem &
        (" " + anv_mruattrib.is_menuitemname, 0)
...
```

of_GetItem

Description

Retrieves MRU information from the DataStore.

Access

Public

Syntax

applicationinstancename.of_GetItem (row, mruinfo)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>row</i>	Long specifying the row to return
<i>mruinfo</i>	N_cst_mruattrib instance into which the function places MRU 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_Restore function:

```
...
    DO UNTIL (li_index >= li_rowcount) OR &
        (li_itemnumber >= li_itemcount)
        li_index++
        of_GetItem(li_index, lnv_mruattrib)
...

```

of_GetRegistered

Description

Retrieves the index of the specified ID within the inv_register array.

Access

Public

Syntax`applicationinstancename.of_GetRegistered(id)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>id</i>	String specifying the ID to be checked

Return value

Integer. Returns the index of *id* within the inv_registered array if *id* is registered and -1 if it is not.

Usage

Internal.

Examples

This example is from the of_IsRegistered function:

```
...
if of_GetRegistered(as_id) < 1 then
    return false
...
...
```

of.GetUserIniFile**Description**

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

Access

Public

Syntax`applicationinstancename.of.GetUserIniFile()`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)

Return value

String. Returns the name of the user-specific INI file.

Examples

This example calls the of.GetUserIniFile function:

```
String ls_file

ls_file = gnv_app.inv_mru.of.GetUserINIFile()
...
...
```

of.GetUserKey**Description**

Retrieves the name of the registry key in which MRU information is stored.

Access

Public

Syntax	<code>applicationinstancename.of_GetUserKey()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>application</i></td><td>Global variable of type n_cst_appmanager (must be gnv_app)</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)</td></tr> </tbody> </table>	Argument	Description	<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)	<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
Argument	Description						
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)						
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)						
Return value	String. Returns the name of the registry key.						
Examples	This example calls the of_GetUserKey function:						
	<pre>String ls_key ls_key = gnv_app.inv_mru.of_GetUserKey() ...</pre>						

of_IsRegistered

Description Reports whether a specified ID is registered with the MRU service.

Access Public

Syntax `applicationinstancename.of_IsRegistered(id)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>id</i>	String specifying the ID to be tested

Return value Boolean. Returns TRUE if *id* is registered and FALSE if it is not.

Examples This example calls the of_IsRegistered function:

```
...
IF NOT gnv_app.inv_mru.of_IsRegistered("myapp") THEN
  gnv_app.inv_mru.of_Register("myapp")
...
```

of_Load

Description Copies MRU information from the registry or INI file to the ids_mrukeys DataStore.

Access Protected

Syntax *application.instancename.of_Load ()*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)

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

Usage Internal.

Examples This example is from the pfc_Open event:

```
IF of_Load() < 1 THEN
    IF IsValid(gnv_app.inv_debug) THEN
        ...
    END IF
END IF
```

of_Read

Description Retrieves MRU items from a specified registry or INI file, calling the pfc_Decode event to extract values.

Access Protected

Syntax *application.instancename.of_Read (section, id)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>section</i>	String specifying the registry or INI file section
<i>id</i>	String specifying the ID to read

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

Usage Internal.

Examples This example is from the of_Load function:

```
...
li_itemcount = UpperBound(istr_register)
FOR li_index = 1 TO li_itemcount
    ls_id = inv_register[li_index].is_id
    li_rc = of_Read(is_mrusection, ls_id)
...
END FOR
```

of_ReadKeys

Description Retrieves values from the registry or INI file.

Access Protected

Syntax *application.instancename.of_ReadKeys (section, keyword, value, default)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>section</i>	String specifying the registry or INI file section
<i>keyword</i>	String specifying the keyword whose corresponding value is to be returned
<i>value</i>	String into which the function places the value corresponding to <i>keyword</i> (passed by reference)
<i>default</i>	String specifying a default value to be returned if <i>keyword</i> is not found

Return value Integer. Returns 1 if the function succeeds, 0 if the default value was used, and -1 if an error occurs.

Usage Internal.

Examples This example is from the of_Load function:

```

...
ls_defaultcount = String(ii_mruitemcount)
of_ReadKeys(is_mrusection, is_mruccountkey, &
            ls_value, ls_defaultcount)
...

```

of_Register

Description Registers an ID to be tracked by the MRU service, optionally specifying the maximum number of items to track.

Access Public

Syntax *application.instancename.of_Register (id {, itemcount })*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>id</i>	String specifying the ID to be registered

	Argument	Description
	<i>itemcount</i> (optional)	Integer specifying the number of MRU items to track. The default is five, which you can modify using the <code>of_SetItemCount</code> function
Return value		Integer. Returns 1 if the function succeeds, 0 if the ID is already registered, and -1 if an error occurs.
Usage		If you specify a number greater than five for <i>itemcount</i> , you must also create additional MRU menu items on the <code>m_master</code> File menu. You must also add code to the window's <code>pfc_PreMRUSave</code> and <code>pfc_MRURestore</code> events that uses the <i>id</i> value. Call this function once for each separate group of MRU items to be tracked. For example, the sheets that display employee information might display different MRU items than the sheets that display customer information.
Examples		This example calls the <code>of_Register</code> function: <code>gnv_app.inv_mru.of_Register("empinfo")</code>

of_Reset

Description	Clears either rows for a specified ID or all rows from the <code>ids_mrukeys</code> DataStore.	
Access	Public	
Syntax	<code>applicationinstancename.of_Reset({id})</code>	
	Argument	Description
	<i>application</i>	Global variable of type <code>n_cst_appmanager</code> (must be <code>gnv_app</code>)
	<i>instancename</i>	Instance name of <code>n_cst_mru</code> (the <code>n_cst_appmanager</code> default is <code>inv_mru</code>)
	<i>id</i> (optional)	String specifying the ID of the MRU items to remove from the <code>ids_mrukeys</code> DataStore
Return value		Integer. Returns the number of rows removed if the function succeeds and -1 if an error occurs.
Examples	This example calls the <code>of_Reset</code> function: ... <code>gnv_app.inv_mru.of_Reset()</code> ...	

of_Restore

Description Adds MRU information to the menu associated with the passed window.

Access Public

Syntax *application.instancename.of_Restore (id, window)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>id</i>	String specifying the id of the MRU items to be restored
<i>window</i>	Window variable referencing the window whose menu is to contain the MRU items

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

Usage The code you add when extending the w_master pfc_MRURestore event calls this function.

Examples This example is from the w_master pfc_MRURestore event:

```
IF IsValid(gnv_app.inv_mru) THEN
    Return gnv_app.inv_mru.of_Restore("empinfo", This)
END IF
```

of_Save

Description Copies information from the ids_mrukeys DataStore to the registry or INI file.

Access Protected

Syntax *application.instancename.of_Save ()*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)

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

Usage Internal.

Examples This example is from the pfc_Close event:

```
If of_Save( ) < 1 Then
    IF IsValid(gnv_app.inv_debug) THEN
        ...
    End If
```

of_SetDelimiter

Description Specifies the delimiter that separates the elements of the MRU value stored in the registry of INI file.

Access Public

Syntax `applicationinstancename.of_SetDelimiter (delimiter)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>delimiter</i>	String specifying the delimiter

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

Usage The default delimiter is the semicolon (;).

Examples This example calls the of_SetDelimiter function:

```
gnv_app.inv_mru.of_SetDelimiter( " ^ " )
```

of_SetItemCount

Description Specifies the maximum number of items that n_cst_mru will track and display at the bottom of the File menu.

Access Public

Syntax `applicationinstancename.of_SetItemCount (count)`

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>count</i>	Integer specifying the maximum number of MRU items

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

Usage If you specify a number larger than five, you must also create additional MRU menu items for menus that use the MRU service.

Examples This example calls the of_SetItemCount function:

```
gnv_app.inv_mru.of_SetItemCount( 4 )
```

of_SetUseRegistry

Description Specifies whether n_cst_mru stores settings in the registry or an INI file.

Access Public

Syntax *application.instancename.of_SetUseRegistry (boolean)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>boolean</i>	Boolean indicating whether the service stores settings in the registry (TRUE) or an INI file (FALSE)

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

Usage Call this function to toggle back and forth between an INI file and the registry (the of_SetUserIniFile and of_SetUserKey functions already control whether the service saves to the INI file or the registry).

The registry is available on Windows platforms only.

Examples This example calls the of_SetUseRegistry function:

```

...
IF ib_restricted THEN
    gnv_app.inv_mru.of_SetUseRegistry(TRUE)
ELSE
    gnv_app.inv_mru.of_SetUseRegistry(FALSE)
END IF
...

```

of_SetUserIniFile

Description Specifies the name of the INI file for MRU information.

Access Public

Syntax *application.instancename.of_SetUserIniFile (inifilename)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>inifilename</i>	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 When you call this function, n_cst_mru uses the specified INI file automatically, overriding any previous specifications made with either the of_SetUseRegistry or of_SetUserKey functions.

The INI file for MRU information is typically the same as the application manager user INI file.

Examples This example from an n_cst_appmanager Constructor event calls the of_SetUserIniFile function:

```
...
IF this.of_IsRegistryAvailable() THEN
    this.inv_mru.of_SetUserKey &
        (this.of_GetUserKey())
ELSE
    this.inv_mru.of_SetUserINIFile &
        (this.of_GetUserINIFile())
END IF
```

of_SetUserKey

Description Specifies the name of the registry key for MRU information.

Access Public

Syntax *applicationinstancename.of_SetUserKey (registrykey)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>registrykey</i>	String containing the registry key

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

Usage When you call this function, n_cst_mru uses the specified registry key automatically, overriding any previous specifications made with either the of_SetUseRegistry or of_SetUserIniFile functions.

The registry key for MRU information is typically the same as the application manager user key.

Examples

This example calls the of_SetUserKey function:

```
...
IF this.of_IsRegistryAvailable() THEN
    this.inv_mru.of_SetUserKey &
        (this.of_GetUserKey())
ELSE
    this.inv_mru.of_SetUserINIFile &
        (this.of_GetUserINIFile())
END IF
```

of_Sort**Description**

Sorts the ids_mrukeys DataStore.

Access

Protected

Syntax

applicationinstancename.of_Sort (sortcriteria)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>sortcriteria</i>	String containing the sort criteria

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:

```
...
of_Filter( " " )
of_Sort( "n_identity D" )
...
```

of_UnRegister**Description**

Removes an ID from the MRU service.

Access

Public

Syntax

applicationinstancename.of_UnRegister (id)

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)

Argument	Description
<i>id</i>	String containing the id to be removed

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

Examples This example calls the of_UnRegister function:

```
gmv_app.inv_mru.of_UnRegister( "empinfo" )
```

of_UpdateVisuals

Description Sets and displays the specified MRU items on the menu associated with the specified window.

Access Protected

Syntax *applicationinstancename.of_UpdateVisuals (window, menuitemtext, menuitemindex)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>window</i>	Window variable referencing the window whose associated menu is updated
<i>menuitemtext</i>	String specifying the text to display on the menu item
<i>menuitemindex</i>	Integer specifying which of the MRU menu items to update. This value must be between 1 and 5

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:

```
...
li_rc = of_UpdateVisuals(aw_window, &
    ls_menuitemtext, li_itemnumber)
...
```

of_Write

Description Copies MRU information from the ids_mrukeys DataStore to the registry or INI file.

Access Protected

Syntax *applicationinstancename.of_Write (section, itemcount)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>section</i>	String specifying the registry or INI file section
<i>itemcount</i>	Integer specifying the number of items to copy

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_rc = of_Write(is_mrusection + " " + ls_id,  &
    inv_register[li_index].ii_itemcount)
...
```

of_WriteKeys

Description Writes information to the registry or INI file.

Access Protected

Syntax *application.instancename.of_WriteKeys (section, keyword, value)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_mru (the n_cst_appmanager default is inv_mru)
<i>section</i>	String specifying the registry or INI file section
<i>keyword</i>	String specifying the keyword whose corresponding value is to be saved
<i>value</i>	String specifying a value corresponding to <i>keyword</i>

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:

```
...
ls_MRUKeyName = is_mruiitemkey +  &
    string(ll_registeredcount)
li_rc = of_WriteKeys(is_mruregisteredsection,  &
    ls_MRUKeyName, ls_key)
...
```

n_cst_nodebase

Description	List instance ancestor object. Implement list nodes through one of this object's descendants:
	<ul style="list-style-type: none"> • N_cst_linkedlistnode • N_cst_treenode
	Each element in a linked list or balanced binary tree is an instance of n_cst_nodebase. n_cst_nodebase contains references to the next and prior nodes, data, a key value, and (balanced binary tree only) balance information.
Ancestry	<pre> graph LR pfc_n_base[pfc_n_base] --> pfc_n_cst_nodebase[pfc_n_cst_nodebase] subgraph pfc_n_base [pfc_n_base] n_base[n_base] Object_functions[Object functions] Protected["Protected of_MessageBox"] end subgraph pfc_n_cst_nodebase [pfc_n_cst_nodebase] n_cst_nodebase[n_cst_nodebase] Object_functions[Object functions] Public["Public of_CopyTo, of_SetData, of_SetKey, of_SetNext, of_SetPrev"] Protected["Protected of_GetData, of_GetKey, of_GetNext, of_GetPrev"] end </pre>
Library	PFCAPSRV.PBL PFEAPSRV.PBL
Usage	PFC objects use this object as the ancestor for list nodes.
Descendants	n_cst_linkedlistnode n_cst_treenode
See also	n_cst_linkedlistbase n_cst_nodecomparebase

Instance variables

N_cst_nodebase includes instance variables:

Instance variable	Description	Data type	Access	Usage
ia_data	Node data	Any	Protected	Set with of_SetData
ia_key	Node key value	Any	Protected	Set with of_SetKey
inv_next	Reference to next node	n_cst_nodebase	Protected	Set with of_SetNext
inv_prev	Reference to prior node	n_cst_nodebase	Protected	Set with of_SetPrev

Functions

N_cst_nodebase contains precoded object functions:

of_CopyTo	of_SetData
of_GetData	of_SetKey
of_GetKey	of_SetNext
of_GetNext	of_SetPrev
of_GetPrev	

of_CopyTo

Description

Copies the data, key, next pointer, and previous pointer of the current node into the passed node.

Access

Public

Syntax

instancename.of_CopyTo (targetnode)

Argument	Description
<i>instancename</i>	Instance name of n_cst_nodebase
<i>targetnode</i>	N_cst_nodebase instance whose information is copied from <i>instancename</i>

Return value

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

Usage

If you extend n_cst_nodebase, be sure to override this function to copy all added data elements as well as the ia_key, ia_data, inv_next, and inv_prev instance variables.

Examples

This example calls the of_CopyTo function:

```
n_cst_linkedlistnode  lnv_node, lnv_temp, lnv_copy
Integer   li_return
Any    la_data

lnv_temp = CREATE n_cst_linkedlistnode
...
lnv_copy = CREATE n_cst_linkedlistnode
li_return = lnv_node.of_CopyTo(lnv_copy)
IF li_return = 1 THEN
    lnv_copy.of_GetData(la_data)
    MessageBox("Success",  &
               "Node copied successfully~r~n"  &
               + "Data is: " + String(la_data))
END IF
...
```

of.GetData

Description Retrieves the data for the current node.

Access Public

Syntax *instancename.of.GetData (data)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_nodebase
<i>data</i>	Any variable into which the function places the data from the current node (passed by reference)

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

Examples This example calls the of.GetData function:

```
...
FOR li_loop = 1 TO li_count
    lnv_node[li_loop].of.GetData(la_data)
    ls_data = String(la_data)
    lb_list.AddItem(ls_data)
NEXT
```

of.GetKey

Description Retrieves the key for the current node.

Access Public

Syntax *instancename.of.GetKey (key)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_nodebase
<i>key</i>	Any variable into which the function places the key value from the current node (passed by reference)

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

Examples This example calls the of.GetKey function:

```
...
FOR li_loop = 1 TO li_count
    lnv_node[li_loop].of.GetData(la_data)
    ls_data = String(la_data)
    lnv_node[li_loop].of.GetKey(la_key)
    ls_key = String(la_key)
    lb_list.AddItem(ls_key + ", " + ls_data)
NEXT
```

of_GetNext

Description Retrieves the next node in the list.

Access Public

Syntax *instancename.of_GetNext (nextnode)*

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_nodebase</i>
<i>nextnode</i>	<i>N_cst_nodebase</i> instance into which the function places the next node in the list (passed by reference)

Return value Integer. Returns 1 if the function succeeds and -1 if an error occurs or there is no next node.

Examples This example is from the *n_cst_list* of *_Find* function:

```
...
lnv_node.of_GetNext(lnv_node)
...
```

of_GetPrev

Description Retrieves the previous node in the list.

Access Public

Syntax *instancename.of_GetPrev (previousnode)*

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_nodebase</i>
<i>previousnode</i>	<i>N_cst_nodebase</i> instance into which the function places the previous node in the list (passed by reference)

Return value Integer. Returns 1 if the function succeeds and -1 if an error occurs or there is no previous node.

Examples This example is from the *n_cst_linkedlistbase* of *_Remove* function:

```
...
any_removenode.of_GetPrev(lnv_previous)
...
```

of_SetData

Description Adds data to the current node.

Access Public

Syntax	<code>instancename.of_SetData (data)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_nodebase
<i>data</i>	Any variable containing the data to be added to <i>instancename</i>

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

Examples This example calls the of_SetData function:

```
n_cst_linkedlistnode lnv_node
Integer li_return

lnv_node = CREATE n_cst_linkedlistnode

lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_list.of_Add(lnv_node)
```

of_SetKey

Description Specifies a key for the current node.

Access Public

Syntax `instancename.of_SetKey (key)`

Argument	Description
<i>instancename</i>	Instance name of n_cst_nodebase
<i>key</i>	Any variable containing the key to be added to <i>instancename</i>

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

Examples This example calls the of_SetKey function:

```
n_cst_linkedlistnode lnv_node
Integer li_return

lnv_node = CREATE n_cst_linkedlistnode

lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_list.of_Add(lnv_node)
```

of_SetNext

Description Establishes a reference to the next node in a list.

Access Public

Syntax *instancename.of_SetNext (nextnode)*

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_nodebase</i>
<i>nextnode</i>	<i>N_cst_nodebase</i> instance that references the new next node in the list

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

Examples This example is from the *n_cst_linkedlistbase* of_Add function:

```
...
    anv_insertafternode.of_SetNext(anv_newnode)
    anv_newnode.of_SetPrev(anv_insertafternode)
...

```

of_SetPrev

Description Establishes a reference to the previous node in a list.

Access Public

Syntax *instancename.of_SetPrev (previousnode)*

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_nodebase</i>
<i>previousnode</i>	<i>N_cst_nodebase</i> instance that references the previous node in the list

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

Examples This example is from the *n_cst_list* of_Add function:

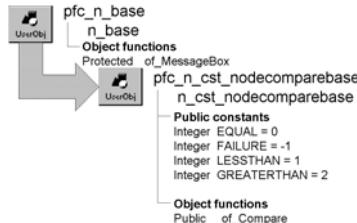
```
...
    anv_insertafternode.of_SetNext(anv_newnode)
    anv_newnode.of_SetPrev(anv_insertafternode)
...

```

n_cst_nodecomparebase

Description Node comparison ancestor object.

Ancestry



Library PFCAPSrv.PBL

PFEAPSrv.PBL

Object relationships n_cst_nodebase

Usage Extend this object's descendants to implement site-specific comparison processing. PFC objects use comparison processing during sort and find operations.

Descendants n_cst_linkedlistnodecompare
n_cst_treenodecompare

See also n_cst_list
n_cst_queue
n_cst_stack
n_cst_tree

Instance variables

N_cst_nodecomparebase includes instance variables:

Instance variable	Description	Data type	Access	Usage
EQUAL	Constant set to 0	Integer	Public	Use with of_Compare
GREATERTHAN	Constant set to 2	Integer	Public	Use with of_Compare
LESSTHAN	Constant set to 1	Integer	Public	Use with of_Compare

Functions

N_cst_nodecomparebase contains a precoded object function:
of_Compare

of_Compare

Compares two items. There are two syntaxes.

To	Use
Compare two nodes	Syntax 1
Compare two values	Syntax 2

Syntax 1

Description

To compare two nodes

Performs a simple comparison using the keys of two passed nodes. Both keys must use a simple data type and the data types must match.

Access

Public

Syntax

instancename.of_Compare (node1, node2)

Argument	Description
<i>instancename</i>	Instance name of n_cst_nodecomparebase
<i>node1</i>	First n_cst_nodebase instance
<i>node2</i>	Second n_cst_nodebase instance

Return value

Integer. Returns values as follows:

- **GREATERTHAN or 2** The key of *node1* is greater than the key of *node2*
- **LESSTHAN or 1** The key of *node1* is less than the key of *node2*
- **EQUAL or 0** The key of *node1* is equal to the key of *node2*
- **FAILURE or -1** An error occurred

Usage

PFC list processing objects call this function when comparing nodes to determine where to place them in the list.

If your list processing requires more complex comparison criteria, create a descendant of either n_cst_linkedlistnodecompare or n_cst_treenodecompare, override the of_Compare function with the appropriate comparison, and establish your customized descendant as the comparison object by calling the list object's of_SetCompare function.

For an example of the code you might use when overriding of_Compare, see n_cst_linkedlistnodecompare on page 1026.

Examples

This example is from the n_cst_list of _Find function:

```

...
do while IsValid(lnv_node)
    choose case inv_compare.of_compare  &
        (anv_keynode, lnv_node)
    case inv_compare.EQUAL
...

```

Syntax 2**To compare two values**

Description

Performs a simple comparison of two passed values.

Access

Public

Syntax

instancename.of_Compare (value1, value2)

Argument	Description
<i>instancename</i>	Instance name of n_cst_nodecomparebase
<i>value1</i>	Any containing the first value
<i>value2</i>	Any containing the second value

Return value

Integer. Returns values as follows:

- **GREATERTHAN or 2** The key of *value1* is greater than the key of *value2*
- **LESSTHAN or 1** The key of *value1* is less than the key of *value2*
- **EQUAL or 0** The key of *value1* is equal to the key of *value2*
- **FAILURE or -1** An error occurred

Usage

Syntax 1 calls this version to perform the actual comparison. The data types for *value1* and *value2* must match.

Examples

This example is from Syntax 1 of the of_Compare function:

```

...
anv_node1.of_getkey(la_key1)
anv_node2.of_getkey(la_key2)

li_rc = of_Compare(la_key1, la_key2)

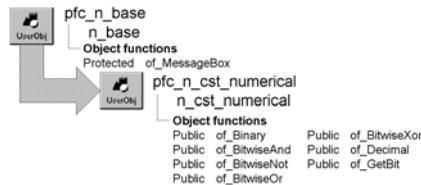
Return li_rc

```

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 do not 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	<i>instancename.of_Binary (number)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_numerical
<i>number</i>	Long to be converted to binary. This number must be positive
Return value	String. Retrieves the binary representation of <i>number</i> . If <i>number</i> 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	<i>instancename.of_BitwiseAnd (value1, value2)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_numerical
<i>value1</i>	Long containing one of the values used in the bitwise AND operation
<i>value2</i>	Long containing the other value used in the bitwise AND operation
Return value	Long. Returns the result of the AND operation on each of <i>value1</i> 's and <i>value2</i> 's bits.
Examples	This example is from the n_cst_filesv of_IncludeFile function:
	<pre> n_cst_numerical lnv_numeric ... IF lnv_numeric.of_BitwiseAnd & (aul_FileAttrib, al_AttribMask) > 0 THEN Return TRUE END IF </pre>

of_BitwiseNot

Description	Performs a bitwise NOT operation on each bit of a passed value. In a bitwise NOT operation:
	<p>1 evaluates to 0 0 evaluates to 1</p>
Access	Public
Syntax	<i>instancename.of_BitwiseNot (value)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_numerical
<i>value</i>	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 <i>value</i> 's bits.
Examples	This example calls the of_BitwiseNot function. It assumes an inv_numerical instance variable:
	<pre> Long ll_original, ll_not ll_original = Long(sle_not.text) ll_not = inv_numerical.of_BitwiseNot & (ll_original) </pre>

```
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
<i>instancename</i>	Instance name of n_cst_numerical
<i>value1</i>	Long containing one of the values used in the bitwise OR operation
<i>value2</i>	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
<i>instancename</i>	Instance name of n_cst_numerical
<i>value1</i>	Long containing one of the values used in the bitwise XOR operation
<i>value2</i>	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
<i>instancename</i>	Instance name of n_cst_numerical
<i>binaryvalue</i>	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
<code>instancename</code>	Instance name of <code>n_cst_numerical</code>
<code>value</code>	Long whose on/off value is to be determined
<code>bit</code>	UnsignedInteger indicating the bit position in <code>value</code> , 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_filesrwin32` `of_DirList` function:

```
n_cst_numerical    lnv_numeric
Integer    li_entries
os_finddata lstr_finddata
...
// Set file attributes
anv_dirlist[li_entries].ib_READONLY = &
    lnv_numeric.of_GetBit &
    (lstr_finddata.ul_fileattributes, 1)
...
```

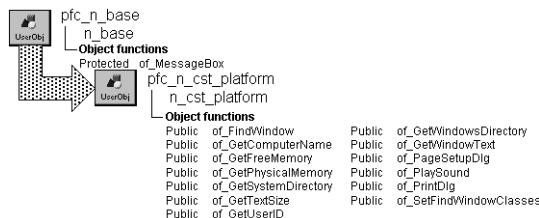
n_cst_platform

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.

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_platformwin32

See also

f_SetPlatform
n_cst_filesvr

Instance variables

`N_cst_platform` contains an instance variable:

Instance variable	Description	Data type	Access	Usage
<code>is_separator</code>	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:

<code>of_FindWindow</code>	<code>of_GetWindowsDirectory</code>
<code>of_GetComputerName</code>	<code>of_GetWindowText</code>
<code>of_GetFreeMemory</code>	<code>of_PageSetupDlg</code>
<code>of_GetPhysicalMemory</code>	<code>of_PlaySound</code>
<code>of_GetSystemDirectory</code>	<code>of_PrintDlg</code>
<code>of_GetTextSize</code>	<code>of_SetFindWindowClasses</code>
<code>of_GetUserID</code>	

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
<code>instancename</code>	Instance name of <code>n_cst_platform</code>
<code>windowname</code>	String

Return value UnsignedInteger. Always returns 0.

See also [n_cst_platformwin32 of_FindWindow function](#)
 [of_SetFindWindowClasses](#)

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
<i>instancename</i>	Instance name of <i>n_cst_platform</i>

Return value String. Always returns an empty string.

See also [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
<i>instancename</i>	Instance name of <i>n_cst_platform</i>

Return value Long. Always returns -1.

See also [n_cst_platformwin32 of_GetFreeMemory 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
<i>instancename</i>	Instance name of n_cst_platform

Return value Long. Always returns -1.

See also 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
<i>instancename</i>	Instance name of n_cst_platform

Return value String. Always returns an empty string.

See also 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
<i>instancename</i>	Instance name of n_cst_platform
<i>window</i>	Window
<i>text</i>	String
<i>font</i>	String
<i>fontsize</i>	Integer
<i>bold</i>	Boolean
<i>italic</i>	Boolean
<i>underline</i>	Boolean

Argument	Description
<i>height</i>	Integer (passed by reference)
<i>width</i>	Integer (passed by reference)

Return value Integer. Always returns -1.

See also 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
<i>instancename</i>	Instance name of n_cst_platform

Return value String. Always returns an empty string.

See also 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
<i>instancename</i>	Instance name of n_cst_platform

Return value String. Always returns an empty string.

See also 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
<i>instancename</i>	Instance name of n_cst_platform
<i>window</i>	UnsignedInteger

Return value String. Always returns an empty string.

See also n_cst_platformwin32 of_GetWindowText 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
<i>instancename</i>	Instance name of n_cst_platform
<i>setupinfo</i>	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
<i>instancename</i>	Instance name of n_cst_platform
<i>filename</i>	String

Argument	Description
<i>beeponfailure</i>	Boolean

Return value Integer. Always returns -1.

See also 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
<i>instancename</i>	Instance name of n_cst_platform
<i>printinfo</i>	S_printDlgattrib instance into which the function places print information (passed by reference)
<i>window</i>	Window

Return value Integer. Always returns -1.

See also n_cst_platformwin32 of _PrintDlg function

of_SetFindWindowClasses

Description Sets the window class names to be used by the service.

Access Public

Syntax *instancename.of_SetFindWindowClasses (classname)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_platform
<i>classname</i>	String name for array of window classes.

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

Usage Of_FindWindow will use all of the class names specified in the *classname* array.

See also of_FindWindow

n_cst_platformwin32

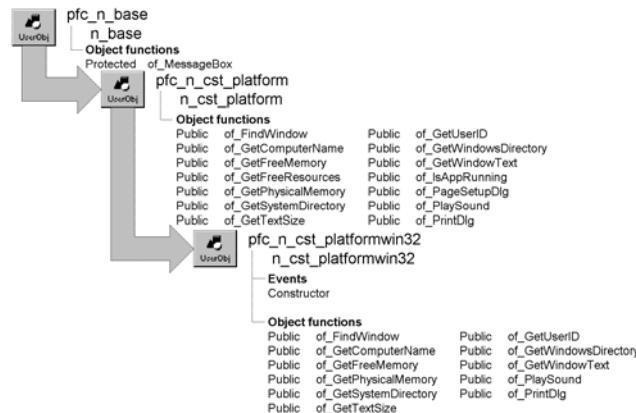
Description

Platform-specific functions for applications running on Windows. 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
- 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](#)

Events

N_cst_platformwin32 includes a pre-coded 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
<i>instancename</i>	Instance name of n_cst_platform
<i>windowname</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_platform
<i>window</i>	Window variable used by the function to create temporary text

Argument	Description
<i>text</i>	String specifying the text for which the function calculates height and width
<i>font</i>	String specifying the font
<i>fontsize</i>	Integer specifying the font size
<i>bold</i>	Boolean specifying whether <i>text</i> is bold
<i>italic</i>	Boolean specifying whether <i>text</i> is italic
<i>underline</i>	Boolean specifying whether <i>text</i> is underlined
<i>height</i>	Integer into which the function places the text height, in PBUs (passed by reference)
<i>width</i>	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_fontsiz, &
    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 |
|---------------------|---------------------------------|
| <i>instancename</i> | 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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_platform
<i>window</i>	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
<i>instancename</i>	Instance name of n_cst_platform
<i>filename</i>	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
<i>instancename</i>	Instance name of n_cst_platform
<i>printinfo</i>	S_printdlgattrib instance containing print information (passed by reference)
<i>window</i>	Window variable containing the window opening the Print dialog box

Return value

Integer. Returns values as follows:

- 0—User cancelled print dialog
- 1—Success
- 1—Unspecified error
- 2—Max pages less than To pages
- 3—Min pages greater than From pages
- 4—From pages or To pages less than 0

Usage

In order to set the Pages radio button, you must specify From and To pages. Pages must be set to display the Collate checkbox.

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)
...

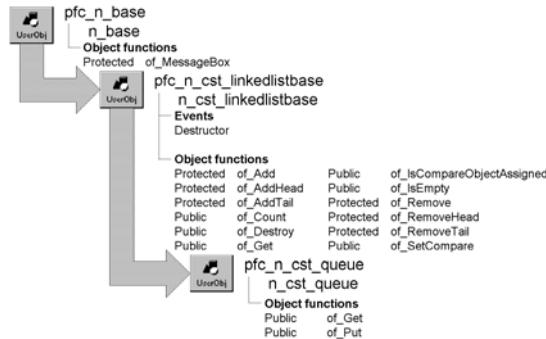
```

n_cst_queue

Description

Queue-processing object. Nodes in a queue are accessed on a first-in, first-out (FIFO) basis. When you add a new node to the queue, n_cst_queue places it at the end of the list; when you get a node from the queue, n_cst_queue accesses it from the beginning of the list, removing it in the process.

Ancestry



Library

PFCAPSRV.PBL

PFEAPSRV.PBL

Object relationships

n_cst_linkedListnode

Usage

Use this object to handle list processing for FIFO lists (queues).

To use n_cst_queue:

- 1 Create a variable for n_cst_queue. This example declares a window instance variable:

```
n_cst_queue  inv_queue
```

- 2 Add nodes to the queue. To do this, first create a node, specify a key and data, then call the of_Put function:

```
n_cst_linkedListnode  lInv_node
Integer  li_return

lInv_node = CREATE n_cst_linkedListnode

lInv_node.of_SetKey(sle_1.text)
lInv_node.of_SetData(sle_1.text)
li_return = inv_queue.of_Put(lInv_node)
```

- 3 Access and remove items from the queue as appropriate by calling the `of_Get` function. This example displays the key in a MessageBox:

```

Any la_key
n_cst_linkedlistnode lnv_node

inv_queue.of_Get(lnv_node)
IF IsValid(lnv_node) THEN
    lnv_node.of_GetKey(la_key)
    MessageBox( "Queue" , &
        "Key is " + String(la_key))
ELSE
    MessageBox( "Queue" , "List is empty")
END IF

```

See also	n_cst_linkedlistnode n_cst_list n_cst_stack n_cst_tree
----------	---

Functions

`N_cst_queue` contains precoded object functions:

`of_Get`
`of_Put`

of_Get

Description Retrieves the top node in the queue, removing it from the queue.

Access Public

Syntax `instancename.of_Get (node)`

Argument	Description
<code>instancename</code>	Instance name of <code>n_cst_queue</code>
<code>node</code>	<code>N_cst_linkedlistnode</code> instance into which the function places a reference to the first node in the queue (passed by reference)

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

Examples

This example calls the of_Get function:

```
Any la_key
n_cst_linkedlistnode lnv_node

inv_queue.of_Get(lnv_node)
IF IsValid(lnv_node) THEN
    lnv_node.of.GetKey(la_key)
    MessageBox("Queue", &
        "Key is " + String(la_key))
ELSE
    MessageBox("Queue", "List is empty")
END IF
```

of_Put**Description**

Adds a node to the bottom of a queue.

Access

Public

Syntax

instancename.of_Put (node)

Argument	Description
<i>instancename</i>	Instance name of n_cst_queue
<i>node</i>	N_cst_linkedlistnode instance to add to the bottom of the queue

Return value

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

Usage

Before calling this function, create the node, add key information, and add data.

Examples

This example calls the of_Put function:

```
n_cst_linkedlistnode lnv_node
Integer li_return

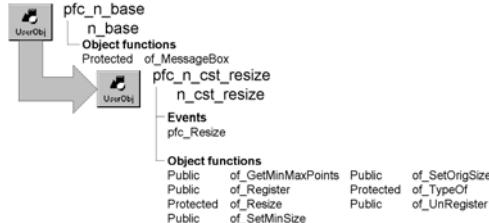
lnv_node = CREATE n_cst_linkedlistnode

lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_queue.of_Put(lnv_node)
```

n_cst_resize

Description Resize service object.

Ancestry



Library PFCWNSRV.PBL

PFEWNSRV.PBL

Usage This service provides two resizing options:

- **For simple resizing** Call the `of_Register` function passing `n_cst_resize` constants, such as `FIXEDBOTTOM`
- **For total control over resizing** Implement weighted resize by calling the `of_Register` function with explicit specifications for moving and scaling

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.

PFC 7 custom class extension

To further the extendibility of PFC, a new *n_cst_resizeattrib* custom class user object replaces the *os_resize* object structure that is now obsolete.

See also

[n_cst_dwsrv_resize](#)
[u_tab](#)
[u_tabpg](#)
[w_master](#)

Instance variables

N_cst_resize includes 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

Instance variable	Description	Data type	Access	Usage
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
inv_registered []	Structure array to track objects eligible for resizing	n_cst_resize attrib	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 pre-coded 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
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>sizetype</i>	UnsignedLong specifying the resize type: <ul style="list-style-type: none">• 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
<i>newwidth</i>	Integer specifying the new width
<i>newheight</i>	Integer specifying the new height

Usage

The u_dw, u_tab, and u_tabpg Resize events call this event.

Functions

N_cst_resize includes pre-coded 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
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>controls</i>	WindowObject array containing the controls for which the minimum and maximum points are returned
<i>minx</i>	Integer into which the function places the minimum x coordinate (passed by reference)
<i>miny</i>	Integer into which the function places the minimum y coordinate (passed by reference)
<i>maxx</i>	Integer into which the function places the maximum x coordinate (passed by reference)
<i>maxy</i>	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

Registers a control for resizing given resize and move percentages.

Access Public

Syntax

instancename.of_Register (control, movex, movey, scalewidth, scaleheight)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_resize</i> (the default is <i>inv_resize</i>)
<i>control</i>	WindowObject variable specifying the object to be resized
<i>movex</i>	Integer specifying the percentage to move the object along the x axis
<i>movey</i>	Integer specifying the percentage to move the object along the y axis
<i>scalewidth</i>	Integer specifying the percentage to resize the object along the x axis
<i>scaleheight</i>	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, 0, 0, 50, 50)
this.inv_resize.of_Register &
(dw_2, 100, 0, 50, 50)
```

Syntax 2

Description Registers a control for resizing given a resize method.

Access Public

Syntax *instancename.of_Register (control, method)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>control</i>	WindowObject variable containing the control to be registered
<i>method</i>	<p>String or n_cst_resize constant specifying how the resize service handles <i>control</i> when the window Resize event occurs:</p> <ul style="list-style-type: none"> • "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
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>newwidth</i>	Integer specifying the new window width
<i>newheight</i>	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.

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

Argument	Description
----------	-------------

<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>minimumwidth</i>	Integer specifying minimum width
<i>minimumheight</i>	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
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>width</i>	Integer specifying the window width
<i>height</i>	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

n_cst_resize

Syntax	<i>instancename.of_TypeOf (control)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>control</i>	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
        ...
    End Case
```

of_UnRegister

Description	Removes a control from the list of controls eligible for resizing.
Access	Public
Syntax	<i>instancename.of_UnRegister (control)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_resize (the default is inv_resize)
<i>control</i>	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:

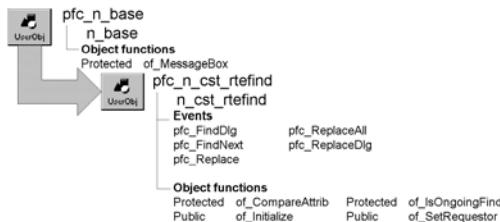
```
Integer li_return

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 <i>n_cst_rtefind</i>	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 <i>n_cst_rtefind</i>	u_rte	Protected	Internal

Events

N_cst_rtefind includes pre-coded 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
<i>instancename</i>	Instance name of n_cst_rtefind
<i>findattrib</i>	N_cst_findattrib instance containing find information. From within the pfc_FindNext event, access this value through the <i>any_findattrib</i> 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
<i>instancename</i>	Instance name of n_cst_rtefind
<i>findattrib</i>	N_cst_findattrib instance containing find information. From within the pfc_Replace event, access this value through the <i>any_findattrib</i> 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
<i>instancename</i>	Instance name of n_cst_rtefind
<i>findattrib</i>	n_cst_findattrib instance containing find information. From within the pfc_ReplaceAll event, access this value through the <i>any_findattrib</i> 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_ReplaceDig

- 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 |
|---------------------|---|
| <i>instancename</i> | Instance name of n_cst_rtefind |
| <i>findattrib</i> | 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	<code>instancename.of_Initialize (findattrib)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_rtefind
<i>findattrib</i>	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	<p>This example calls the of_Initialize function:</p> <pre>// 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)</pre>

of_IsOngoingFind

Description	Reports whether the current find operation is a new Find or Find Next operation.
Access	Public
Syntax	<code>instancename.of_IsOngoingFind (findattrib)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_rtefind
<i>findattrib</i>	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 if 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
<i>instancename</i>	Instance name of n_cst_rtefind
<i>requestor</i>	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
    ...

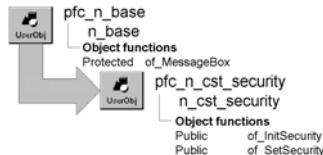
```

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 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 *applicationinstancename.of_InitSecurity (transaction, appname, user, group {, precache })*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>instancename</i>	Instance name of n_cst_security (the n_cst_appmanager default is inv_security)

Argument	Description
<i>transaction</i>	N_tr variable for the security database connection. You must connect to the database before calling this function
<i>appname</i>	String specifying the application name, as defined in the security database
<i>user</i>	String specifying the user ID, as defined in the security database
<i>group</i>	String specifying the default group. PFC uses this group if no group is defined in the database for <i>user</i>
<i>precache</i> (optional)	Boolean specifying whether to preload all security settings for the user (the default is FALSE)
Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none"> • 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	<p>You typically call this function in the n_cst_appmanager pfc_Open event.</p> <p>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.</p>
Examples	<p>This example calls the of_InitSecurity function. It assumes an itr_sec instance variable of type n_tr:</p> <pre> 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") ... </pre>

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 *applicationinstancename.of_SetSecurity (object)*

Argument	Description
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>instancename</i>	Instance name of n_cst_security (the n_cst_appmanager default for this is inv_security)
<i>object</i>	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	 <pre> graph TD pfc_n_base[pfc_n_base] --> n_base[n_base] n_base --> pfc_n_cst_selection[pfc_n_cst_selection] pfc_n_cst_selection --> n_cst_selection[n_cst_selection] </pre> <p>The diagram illustrates the object hierarchy. It starts with 'pfc_n_base' (UserObj) at the top, which has a protected function 'of_MessageBox'. Below it is 'n_base' (UserObj), which has an object function 'Protected of_MessageBox'. An arrow points from 'n_base' to 'pfc_n_cst_selection' (UserObj). 'pfc_n_cst_selection' has a public function 'of_Open'. Below it is 'n_cst_selection' (UserObj), which also has an object function 'Public of_Open'.</p>
Library	PFCAPSRV.PBL PFEAPSRV.PBL
Object relationships	n_tr n_cst_selectionattrib w_selection
Usage	<p>To use n_cst_selection functions:</p> <ol style="list-style-type: none"> 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

Description

To display retrieved rows for a DataWindow

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
<i>instancename</i>	Instance name of n_cst_selection
<i>dataobject</i>	String specifying the DataWindow object that w_selection uses to retrieve and display rows
<i>returnval</i>	Array of the Any data type into which the w_selection dialog box places information for the selected columns (passed by reference)
<i>transobj</i> (optional)	N_tr-based Transaction object specifying an active connection that the w_selection dialog box uses to retrieve rows
<i>columns</i> (optional)	String specifying the DataWindow columns that w_selection returns. The default is the first column
<i>arguments</i> (optional)	Twenty-element array of the Any data type specifying retrieval arguments for <i>dataobject</i>
<i>title</i> (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 <code>w_selection</code> and Syntax 3 to display data stored in a DataWindow object.														
Examples	This example calls the <code>of_Open</code> function:														
	<pre>Any la_values[] String ls_col[] ls_col[1] = "emp_id" inv_selection.of_Open("d_emplist", la_values, & SQLCA, ls_col) ... </pre>														
Syntax 2	To display a passed set of rows														
Description	Displays the <code>w_selection</code> dialog box using rows passed in a function argument. The user selects a one or more columns in <code>w_selection</code> , which returns selection information for the selected columns.														
Access	Public														
Syntax	<code>instancename.of_Open (dataobject, returnval {, powerobj {, columns {, title } } })</code>														
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><code>instancename</code></td><td>Instance name of <code>n_cst_selection</code></td></tr> <tr> <td><code>dataobject</code></td><td>String specifying the DataWindow object that <code>w_selection</code> uses to display rows</td></tr> <tr> <td><code>returnval</code></td><td>Array of the Any data type into which the <code>w_selection</code> dialog box places information for the selected columns (passed by reference)</td></tr> <tr> <td><code>powerobj (optional)</code></td><td>PowerObject array containing data that the <code>w_selection</code> dialog box displays in the <code>dataobject</code> DataWindow</td></tr> <tr> <td><code>columns (optional)</code></td><td>String specifying the DataWindow columns that <code>w_selection</code> returns. The default is the first column</td></tr> <tr> <td><code>title (optional)</code></td><td>String specifying a title for the <code>w_selection</code> dialog box</td></tr> </tbody> </table>	Argument	Description	<code>instancename</code>	Instance name of <code>n_cst_selection</code>	<code>dataobject</code>	String specifying the DataWindow object that <code>w_selection</code> uses to display rows	<code>returnval</code>	Array of the Any data type into which the <code>w_selection</code> dialog box places information for the selected columns (passed by reference)	<code>powerobj (optional)</code>	PowerObject array containing data that the <code>w_selection</code> dialog box displays in the <code>dataobject</code> DataWindow	<code>columns (optional)</code>	String specifying the DataWindow columns that <code>w_selection</code> returns. The default is the first column	<code>title (optional)</code>	String specifying a title for the <code>w_selection</code> dialog box
Argument	Description														
<code>instancename</code>	Instance name of <code>n_cst_selection</code>														
<code>dataobject</code>	String specifying the DataWindow object that <code>w_selection</code> uses to display rows														
<code>returnval</code>	Array of the Any data type into which the <code>w_selection</code> dialog box places information for the selected columns (passed by reference)														
<code>powerobj (optional)</code>	PowerObject array containing data that the <code>w_selection</code> dialog box displays in the <code>dataobject</code> DataWindow														
<code>columns (optional)</code>	String specifying the DataWindow columns that <code>w_selection</code> returns. The default is the first column														
<code>title (optional)</code>	String specifying a title for the <code>w_selection</code> dialog box														
Return value	Integer. Returns the number of elements in the <code>returnval</code> 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_value   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
<i>instancename</i>	Instance name of n_cst_selection
<i>dataobject</i>	String specifying the DataWindow object that w_selection uses to retrieve and display rows
<i>returnval</i>	Array of the Any data type into which the w_selection dialog box places information for the selected columns (passed by reference)
<i>columns (optional)</i>	String specifying the DataWindow columns that w_selection returns. The default is the first column
<i>title (optional)</i>	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

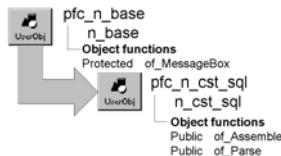
This example calls the of_Open function:

```
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	<i>instancename.of_Assemble (components)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_sql
<i>components</i>	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 <i>components</i> 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:
	<pre> String ls_sql n_cst_sqlattrib lInv_sqlattrib[] lInv_sqlattrib[1].s_verb = sle_verb.text lInv_sqlattrib[1].s_tables = sle_tables.text lInv_sqlattrib[1].s_columns = sle_columns.text lInv_sqlattrib[1].s_values = sle_values.text lInv_sqlattrib[1].s_where = sle_where.text lInv_sqlattrib[1].s_order = sle_order.text lInv_sqlattrib[1].s_group = sle_group.text lInv_sqlattrib[1].s_having = sle_having.text ls_sql = inv_sql.of_Assemble(lInv_sqlattrib) MessageBox("SQL" , ls_sql) </pre>

of_Parse

Description	Separates a SQL statement into its component parts.
Access	Public
Syntax	<i>instancename.of_Parse (statement, components)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_sql
<i>statement</i>	String containing the SQL statement to be separated
<i>components</i>	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 <i>components</i> 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	<p>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.</p> <p>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.</p> <p>For complete information on using SQL Spy, see the <i>PFC User's Guide</i>.</p>
Ancestry	<pre> graph TD pfc_n_base[pfc_n_base] -- "Protected of _MessageBox" --> n_base[n_base] pfc_n_cst_sqlspy[pfc_n_cst_sqlspy] -- "Public of _GetAlwaysOnTop, Public of _OpenSQLSpy, Public of _SetAlwaysOnTop" --> n_cst_sqlspy[n_cst_sqlspy] </pre>
Library	<p>PF</p> <p>PFCUTIL.PBL PFEUTIL.PBL</p>
Object relationships	<p>d_sqlspypoint n_cst_conversion n_ds w_sqlspy w_sqlspyinspect</p>
Usage	<p>You typically call most n_cst_sqlspy functions from within the application manager. To use SQL Spy:</p> <ol style="list-style-type: none"> 1 Enable the debugging service by calling the n_cst_appmanager of_SetDebug function: <pre> this.of_SetDebug(TRUE)</pre> <ol style="list-style-type: none"> 2 Enable SQL Spy by calling the n_cst_debug of_SetSQLSpy function: <pre> this.inv_debug.of_SetSQLSpy(TRUE)</pre> <ol style="list-style-type: none"> 3 Specify the log file by calling the n_cst_sqlspy of_SetLogFile function: <pre> this.inv_debug.inv_sqlspy.of_SetLogFile & ("c:\debug\temp\appdbug.log")</pre> <ol style="list-style-type: none"> 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
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>debug</i>	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
<i>sqlspy</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>debug</i>	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
<i>sqlspy</i>	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
<i>boolean</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>debug</i>	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
<i>sqlspy</i>	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
<i>boolean</i>	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_sqspy.of_SetAlwaysOnTop &  
    (FALSE)  
...
```

of_SetBatchMode

Description

Controls whether the service displays the w_sqspyinspect 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
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>debug</i>	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
<i>sqlspy</i>	Instance name of n_cst_sqspy (the n_cst_debug default is inv_sqspy)
<i>boolean</i>	Boolean indicating whether to display the w_sqspyinspect 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_sqspy.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
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>debug</i>	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
<i>sqlspy</i>	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
<i>logfile</i>	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
<i>application</i>	Global variable of type n_cst_appmanager (this must be gnv_app)
<i>debug</i>	Instance name of n_cst_debug (the n_cst_appmanager default is inv_debug)
<i>sqlspy</i>	Instance name of n_cst_sqlspy (the n_cst_debug default is inv_sqlspy)
<i>heading</i>	String specifying a heading for the log file
<i>sql</i> (optional)	String specifying the SQL statement to be logged
<i>oktopause</i> (optional)	Boolean specifying whether to pause statement execution for display in the w_sqlspyinspect dialog box. The default is FALSE

Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none">• 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	<p>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.</p> <p>To display the statement in the w_sqlspying inspect dialog box, <i>oktopause</i> must be TRUE and batch mode must have been disabled previously by calling the of_SetBatchMode(FALSE) function</p>

Examples

This example calls the of_SQLSyntax function:

```
String    ls_sql

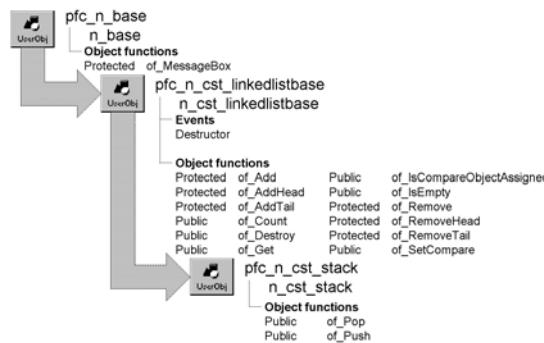
ls_sql = "SELECT * FROM employee;"  
gnv_app.inv_debug.inv_sqlspy.of_SQLSyntax &  
        ("Native SQL", ls_sql, TRUE)  
...
```

n_cst_stack

Description

Stack processing object. Nodes in a stack are accessed on a last-in, first-out (LIFO) basis. When you add a new node to the stack, n_cst_stack places it at the beginning of the list; when you get a node from the stack, n_cst_stack accesses it from the beginning of the list, removing it in the process.

Ancestry



Library

PFCAPSRV.PBL

PFEAPSRV.PBL

Object relationships

n_cst_linkedlistnode

Usage

Use this object to handle list processing for LIFO lists (stacks).

To use n_cst_stack:

- 1 Create an instance of n_cst_stack. This example declares a window instance variable:

```
n_cst_stack inv_stack
```

- 2 Add nodes to the stack. To do this, first create a node, specify a key and data, then call the of_Push function:

```

n_cst_linkedlistnode lnv_node
Integer li_return

lnv_node = CREATE n_cst_linkedlistnode

lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_stack.of_Push(lnv_node)
  
```

- 3 Access and remove items from the queue as appropriate by calling the of_Pop function. This example displays the key in a MessageBox:

```
Any la_key  
n_cst_linkedlistnode lnv_node  
  
inv_stack.of_Pop(lnv_node)  
IF IsValid(lnv_node) THEN  
    lnv_node.of_GetKey(la_key)  
    MessageBox("Stack", &  
              "Key is " + String(la_key))  
ELSE  
    MessageBox("Stack", "Stack is empty")  
END IF
```

See also

[n_cst_linkedlistnodecompare](#)
[n_cst_list](#)
[n_cst_queue](#)
[n_cst_tree](#)

Functions

N_cst_stack includes precoded object functions:

of_Pop
of_Push

of_Pop

Description

Retrieves the top node in the stack, removing it from the stack.

Access

Public

Syntax

instancename.of_Pop (node)

Argument	Description
<i>instancename</i>	Instance name of n_cst_stack
<i>node</i>	N_cst_linkedlistnode instance into which the function places a reference to the top node in the stack (passed by reference)

Return value

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

Examples

This example calls the of_Pop function:

```

Any la_key
n_cst_linkedlistnode lnv_node

inv_stack.of_Pop(lnv_node)
IF IsValid(lnv_node) THEN
    lnv_node.of_GetKey(la_key)
    MessageBox("Stack", &
        "Key is " + String(la_key))
ELSE
    MessageBox("Stack", "Stack is empty")
END IF

```

of_Push**Description**

Adds a node to the top of the stack.

Access

Public

Syntax

instancename.of_Push (node)

Argument	Description
<i>instancename</i>	Instance name of n_cst_stack
<i>node</i>	N_cst_linkedlistnode instance to add to the top of the stack

Return value

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

Usage

Before calling this function, create the node, add key information, and add data.

Examples

This example calls the of_Push function:

```

n_cst_linkedlistnode lnv_node
Integer li_return

lnv_node = CREATE n_cst_linkedlistnode

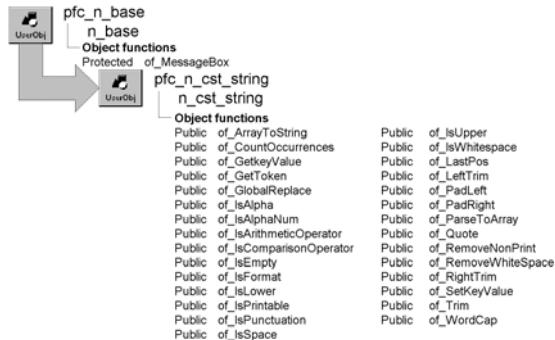
lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_stack.of_Push(lnv_node)

```

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, {processeempty} target)

Argument	Description
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String array to be converted into a single string
<i>delimiter</i>	String specifying the delimiter the function places between array elements
<i>processeempty</i>	Boolean allowing processing of empty string arguments if true
<i>target</i>	String to contain the <i>source</i> 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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	String to be searched
<i>target</i>	String to search for
<i>ignorecase</i> (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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	String containing the complete expression
<i>keyword</i>	String containing the keyword to be searched for
<i>separator</i>	String containing the separator character in <i>source</i>

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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String to be parsed (passed by reference)
<i>separator</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String to be searched
<i>lookfor</i>	String to be found
<i>replacewith</i>	String used to replace <i>lookfor</i>
<i>ignorecase</i> (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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string

Argument	Description
<i>source</i>	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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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_LTrim 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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String containing the complete expression
<i>target</i>	String containing the expression to be searched for
<i>start</i> (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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String to be trimmed
<i>removespaces</i> (optional)	Boolean specifying whether to remove spaces. The default is TRUE
<i>removenonprintable</i> (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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String to be expanded
<i>length</i>	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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	String to be expanded
<i>length</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String to be converted into an array
<i>delimiter</i>	String specifying the character(s) that separate array elements in <i>source</i>
<i>array</i>	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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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	<code>instancename.of_RemoveNonPrint (source)</code>
Argument	Description
<code>instancename</code>	Instance name of n_cst_string
<code>source</code>	String from which the function removes nonprintable characters
Return value	String. Returns <code>source</code> with nonprintable characters removed. If <code>source</code> is NULL, the function returns NULL.
Examples	This example calls the <code>of_RemoveNonPrint</code> 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	<i>instancename.of_RemoveWhiteSpace (source)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String from which the function removes whitespace characters
Return value	String. Returns <i>source</i> with whitespace characters removed. If <i>source</i> 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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	String to be trimmed
<i>removespaces</i> (optional)	Boolean specifying whether to remove spaces. The default is TRUE
<i>removenonprintable</i> (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_LTrim(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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	String whose value is to be set (passed by reference)
<i>keyword</i>	String specifying the keyword whose value will be set
<i>newvalue</i>	String containing the new value
<i>separator</i>	String specifying the separator character in <i>source</i>

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
<i>instancename</i>	Instance name of <i>n_cst_string</i>
<i>source</i>	String to be trimmed
<i>removespaces</i> (optional)	Boolean specifying whether to remove spaces. The default is TRUE

Argument	Description
<i>removenonprintable</i> (optional)	Boolean specifying whether to remove nonprintable characters (for example, newline, tab, and carriage return). The default is FALSE
Return value	String. Returns <i>source</i> 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	<p>This example calls the of_Trim function:</p> <pre> 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 ... </pre>

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
<i>instancename</i>	Instance name of n_cst_string
<i>source</i>	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

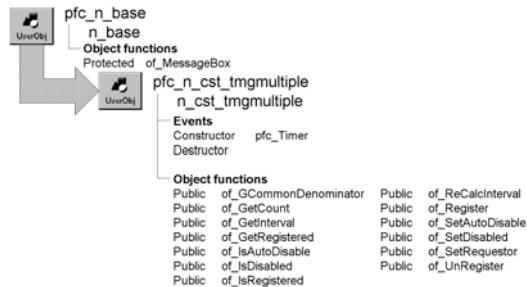
```

n_cst_tmgtmultiple

Description

Timing service (multiple timers). Use this service to maintain an unlimited number of timers triggered at specified intervals (with a one second granularity).

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Object relationships

n_cst_datetime
n_ds
n_tmg
n_tmgregisterattrib

Usage

Use this service to implement multiple timers. To use this service:

- 1 Establish an instance variable of type n_tmg:

```
n_tmg itmg_timer
```

- 2 Create the instance of n_tmg:

```
itmg_timer = CREATE n_tmg
```

- 3 Enable the multiple timer service:

```
itmg_timer.of_SetMultiple(TRUE)
```

- 4 Register the objects and events to be notified (window are the objects in this example):

```
itmg_timer.inv_multiple.of_Register &  
    (this, "ue_timer1", 5)  
itmg_timer.inv_multiple.of_Register &  
    (this, "ue_timer2", 10)  
itmg_timer.inv_multiple.of_Register &  
    (this, "ue_timer3", 30)
```

5 Code the events to receive notification from n_cst_tmgtmultiple (ue_timer1, ue_timer2, and ue_timer3 in this example):

6 (optional) Unregister objects and events:

```
itmg_timer.inv_multiple.of_UnRegister()
```

This service is for use with n_tmg, the PFC timing object.

See also

n_cst_tmgsingle

Instance variables

N_cst_tmgtmultiple includes instance variables:

Instance variable	Description	Data type	Access	Usage
ib_autodisable	Indicates whether the service enables autodisable	Boolean	Protected	Set with of_SetAutoDisable
ib_disabled	Indicates whether a notification can occur	Boolean	Protected	Set with of_SetDisabled
ids_list	DataStore containing information on registered objects	n_ds	Protected	Internal
il_interval	Timer interval	Long	Protected	Access with of_GetInterval
inv_datetime	DateTime service	n_cst_datetime	Protected	Internal
inv_tmgtregister[]	Information on registered objects	n_cst_tmgtregisterattrib	Protected	Internal
itmg_requestor	N_tmg instance that owns the n_cst_tmgtmultiple instance	n_tmg	Protected	Internal
POSTEVENT	Constant set to 1	Integer	Public	Use with of_Register
TRIGGEREVENT	Constant set to 0	Integer	Public	Use with of_Register

Events

N_cst_tmgtmultiple includes precoded events:

Constructor	pfc_Timer
Destructor	

Constructor

Description	Creates an instance of ids_list.
Usage	This event executes when the object is created.

Destructor

Description	Destroys the instance of ids_list.
Usage	This event executes when the object is destroyed.

pfc_Timer

Description	Notifies the registered object that the timer has fired. Depending on how you registered the objects, this event either triggers or posts the specified events.
Usage	The n_tmg Timer event calls this event.

Functions

N_cst_tmgmultiple includes pre-coded object functions:

of_GCommonDenominator	of_RecalcInterval
of_GetCount	of_Register
of_GetInterval	of_SetAutoDisable
of_GetRegistered	of_SetDisabled
of_IsAutoDisable	of_SetRequestor
of_IsDisabled	of_UnRegister
of_IsRegistered	

of_GCommonDenominator

Description	Retrieves the greatest common denominator of two values.
Access	Public
Syntax	<code>tmgcontrolinstancename.of_GCommonDenominator (value1, value2)</code>

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)
<i>value1</i>	Long containing the first value

	Argument	Description
	<i>value2</i>	Long containing the second value
Return value		Long. Returns the greatest common denominator of the two values if the function succeeds and -1 if an error occurs.
Usage		Internal.
Examples		This example is from the of_RecalcInterval function:
		...
		ll_gcd = of_GCommonDenominator &
		(inv_tmregister[li_idx].il_notifyinterval,ll_gcd)
		...

of_GetCount

Description Retrieves the number of registered objects.

Access Public

Syntax *tmgcontrolinstancename.of_GetCount ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)

Return value Integer. Returns the number of registered objects if the function succeeds, and -1 if an error occurs.

Examples This example calls the of_GetCount function:

```
Integer li_return

IF IsValid(itmg_timer.inv_multiple) THEN
    li_return = &
    itmg_timer.inv_multiple.of_GetCount()
END IF
sle_1.text = String(li_return)
```

of_GetInterval

Description Retrieves the current interval.

Access Public

Syntax *tmgcontrolinstancename.of_GetInterval ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmgt-based timing control
<i>instancename</i>	Instance name of n_cst_tmgtmultiple (the n_tmgt default is inv_multiple)

Return value Long. Returns the current interval if the function succeeds and 0 if no interval has been defined.

Examples This example calls the of_GetInterval function:

```
Long ll_interval

IF IsValid(itmg_timer.inv_multiple) THEN
    ll_interval =  &
        itmg_timer.inv_multiple.of_GetInterval()
END IF
sle_1.text = "Interval is: " &
    + String(ll_interval)
```

of_GetRegistered

Description Retrieves information for registered objects.

Access Public

Syntax *tmgcontrol.instancename.of_GetRegistered (objects)*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmgt-based timing control
<i>instancename</i>	Instance name of n_cst_tmgtmultiple (the n_tmgt default is inv_multiple)
<i>objects</i>	N_cst_tmgtregisterattrib array into which the function places information on registered objects (passed by reference)

Return value Integer. Returns the number of entries in the *objects* array if the function succeeds and -1 if an error occurs.

Examples This example calls the of_GetRegistered function:

```
n_cst_tmgtregisterattrib lnv_objects[ ]
Integer li_return

IF IsValid(itmg_timer.inv_multiple) THEN
    li_return =  &
        itmg_timer.inv_multiple.of_GetRegistered  &
        (lnv_objects)
```

```
END IF
...
```

of_IsAutoDisable

Description Reports whether the service is using the autodisable feature.

Access Public

Syntax *tmgcontrolinstancename.of_IsAutoDisable ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgtmultiple (the n_tmg default is inv_multiple)

Return value Boolean. Returns TRUE if autodisable is enabled and FALSE if it is not.

Examples This example is from the pfc_Timer event:

```
...
If of_IsAutoDisable() Then
    of_SetDisabled(False)
End If
...
```

of_IsDisabled

Description Reports whether notification is disabled. If autodisable is enabled, n_cst_tmgtmultiple disables itself automatically while notifying registered objects.

Access Public

Syntax *tmgcontrolinstancename.of_IsDisabled ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgtmultiple (the n_tmg default is inv_multiple)

Return value Boolean. Returns TRUE if notification is disabled and FALSE if it is not.

Examples This example is from the pfc_Timer event:

```
If of_IsDisabled() Then
    Return 0
```

```
End If  
...
```

of_IsRegistered

Description Reports whether the specified object or object/event combination is registered.

Access Public

Syntax *tmgcontrolinstancename.of_IsRegistered (object { , event })*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)
<i>object</i>	PowerObject referencing the object to be tested
<i>event</i> (optional)	String specifying the event defined within <i>object</i>

Return value Boolean. Returns TRUE if *object* or *object* and *event* are registered with the service and FALSE if they are not.

Examples This example calls the of_IsRegistered function:

```
IF IsValid(itmg_timer.inv_multiple) THEN  
    IF  
        itmg_timer.inv_multiple.of_IsRegistered(parent)  
        THEN  
            sle_1.text = "Window is registered"  
        ELSE  
            sle_1.text = "Not registered"  
        END IF  
    END IF
```

of_RecalcInterval

Description Calculates a new interval.

Access Public

Syntax *tmgcontrolinstancename.of_RecalcInterval ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)

Return value Long. Returns the new interval if the function succeeds and -1 if an error occurs.

Usage Internal.

Examples This example is from the pfc_Timer event:

```
...
If lb_deleted Then
    of_ReCalcInterval()
End If
...
```

of_Register

Description Registers an object instance (or object instance and event) with n_cst_tmgtmultiple.

Access Public

Syntax *tmgcontrolinstancename.of_Register (object, event, interval{ , style })*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgtmultiple (the n_tmg default is inv_multiple)
<i>object</i>	PowerObject referencing the instance to be notified when the timer fires
<i>event</i>	String specifying the event to be called. This must be a valid event for <i>object</i>
<i>interval</i>	Long specifying the timer interval in seconds
<i>style</i> (optional)	Integer or n_cst_tmgtmultiple constant specifying the notification style: <ul style="list-style-type: none"> • 0 or TRIGGEREVENT Trigger the event (default) • 1 or POSTEVENT Post the event

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

Usage Call this function to specify the object, event, and interval used by the timer.

If you specify an interval of zero, the service disables the timer.

Examples This example calls the of_Register function:

```
IF IsValid(itmg_timer.inv_multiple) THEN
    itmg_timer.inv_multiple.of_Register  =
        (parent, "ue_showtimer", 15)
END IF
```

of_SetAutoDisable

Description Specifies whether *n_cst_tmgmultiple* disables notifications automatically during the notification process.

Access Public

Syntax *tmgcontrol.instancename.of_SetAutoDisable (boolean)*

Argument	Description
<i>tmgcontrol</i>	Instance name of the <i>n_tm</i> -based timing control
<i>instancename</i>	Instance name of <i>n_cst_tmgmultiple</i> (the <i>n_tm</i> default is <i>inv_multiple</i>)
<i>boolean</i>	Boolean specifying whether the service disables notification automatically (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_SetAutoDisable* function:

```
itmഗ_timer.inv_multiple.of_SetAutoDisable(TRUE)
```

of_SetDisabled

Description Specifies whether *n_cst_tmgmultiple* sends notifications to registered objects.

Access Public

Syntax *tmgcontrol.instancename.of_SetDisabled (boolean)*

Argument	Description
<i>tmgcontrol</i>	Instance name of the <i>n_tm</i> -based timing control
<i>instancename</i>	Instance name of <i>n_cst_tmgmultiple</i> (the <i>n_tm</i> default is <i>inv_multiple</i>)
<i>boolean</i>	Boolean specifying whether to the service sends notifications to registered objects (TRUE) or not (FALSE)

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

Usage If autodisable is enabled, the pfc_Timer event calls this function to prevent simultaneous notifications.

Examples This example is from the pfc_Timer event:

```
If of_IsAutoDisable() Then  
    of_SetDisabled(True)  
End If
```

of_SetRequestor

Description	Associates the requesting object with this instance of n_cst_tmgmultiple.								
Access	Public								
Syntax	<code>tmgcontrol.instancename.of_SetRequestor (requestor)</code>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>tmgcontrol</i></td><td>Instance name of the n_tmg-based timing control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)</td></tr> <tr> <td><i>requestor</i></td><td>N_tmg-based timing object instance to be associated with the service</td></tr> </tbody> </table>	Argument	Description	<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control	<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)	<i>requestor</i>	N_tmg-based timing object instance to be associated with the service
Argument	Description								
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control								
<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)								
<i>requestor</i>	N_tmg-based timing object instance 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 n_tmg of_SetSingle function:

```

...
inv_multiple = Create n_cst_tmgmultiple
inv_multiple.of_SetRequestor(this)
...

```

of_UnRegister

Description	Unregisters the current notification object.										
Access	Public										
Syntax	<code>tmgcontrol.instancename.of_UnRegister ({ object { , event } })</code>										
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>tmgcontrol</i></td><td>Instance name of the n_tmg-based timing control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)</td></tr> <tr> <td><i>object</i> (optional)</td><td>PowerObject specifying the object to unregister. If you don't specify this argument the function unregisters all objects</td></tr> <tr> <td><i>event</i> (optional)</td><td>String specifying the event to unregister. If you don't specify this argument the function unregisters all events for <i>object</i></td></tr> </tbody> </table>	Argument	Description	<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control	<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)	<i>object</i> (optional)	PowerObject specifying the object to unregister. If you don't specify this argument the function unregisters all objects	<i>event</i> (optional)	String specifying the event to unregister. If you don't specify this argument the function unregisters all events for <i>object</i>
Argument	Description										
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control										
<i>instancename</i>	Instance name of n_cst_tmgmultiple (the n_tmg default is inv_multiple)										
<i>object</i> (optional)	PowerObject specifying the object to unregister. If you don't specify this argument the function unregisters all objects										
<i>event</i> (optional)	String specifying the event to unregister. If you don't specify this argument the function unregisters all events for <i>object</i>										

Return value Integer. Returns the number of unregistered entries if the function succeeds, 0 if the entry was not found, and -1 if an error occurs.

Examples This example calls the of_UnRegister function:

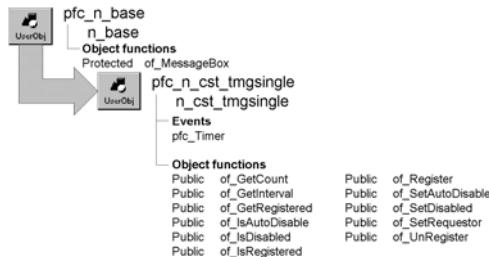
```
itmഗ_timer.inv_multiple.of_UnRegister()
```

n_cst_tmgsingle

Description

Timing service (single timer). Use this service to maintain a single timer, triggered at regular intervals between .55 and 65 seconds.

Ancestry



Library

PFCAPSRV.PBL

PFEAPSRV.PBL

Object relationships

n_tmg

Usage

Use this service to implement a single timer. To use this service:

- 1 Establish an instance variable of type n_tmg:

```
n_tmg itmg_timer
```

- 2 Create the instance of n_tmg:

```
itmg_timer = CREATE n_tmg
```

- 3 Enable the single timer service:

```
itmg_timer.of_SetSingle(TRUE)
```

- 4 Register the object and the event to be notified (the window is the object in this example):

```
IF IsValid(itmg_timer.inv_single) THEN
    itmg_timer.inv_Single.of_Register  &
    (this, "ue_showtimer", 15)
END IF
```

- 5 Code the event to receive notification from n_cst_tmgsingle (ue_showtimer in this example):

- 6 (Optional) Unregister the object and event:

```
itmg_timer.inv_single.of_UnRegister()
```

This service is for use with n_tmg, the PFC timing object.

See also

n_cst_tmgsingle

Instance variables

`N_cst_tmgsingle` includes instance variables:

Instance variable	Description	Data type	Access	Usage
<code>ib_autodisable</code>	Indicates whether the service enables autodisable	Boolean	Protected	Set with <code>of_SetAutoDisable</code>
<code>ib_disabled</code>	Indicates whether a notification can occur	Boolean	Protected	Set with <code>of_SetDisabled</code>
<code>ii_notifystyle</code>	Notification style: <code>TriggerEvent</code> or <code>PostEvent</code>	Integer	Protected	Set with <code>of_Register</code>
<code>ipo_notifyobject</code>	Object to notify when the timer fires	PowerObject	Protected	Set with <code>of_Register</code>
<code>ir_interval</code>	Timer interval	Real	Protected	Set with <code>of_Register</code>
<code>is_notifyevent</code>	Event to notify when the timer fires	String	Protected	Set with <code>of_Register</code>
<code>itmg_requestor</code>	<code>N_tmg</code> instance that owns the <code>n_cst_tmgsingle</code> instance	<code>n_tmg</code>	Protected	Internal
<code>POSTEVENT</code>	Constant set to 1	Integer	Public	Use with <code>of_Register</code>
<code>TRIGGEREVENT</code>	Constant set to 0	Integer	Public	Use with <code>of_Register</code>

Events

`N_cst_tmgsingle` includes a precoded event:

`pfc_Timer`

`pfc_Timer`

Description	Notifies the registered object that the timer has fired. Depending on how you registered the object, this event either triggers or posts the specified event.
Usage	The <code>n_tmg</code> Timer event calls this event.

Functions

`N_cst_tmgsingle` includes precoded object functions:

<code>of_GetCount</code>	<code>of_Register</code>
<code>of_GetInterval</code>	<code>of_SetAutoDisable</code>

of_GetRegistered	of_SetDisabled
of_IsAutoDisable	of_SetRequestor
of_IsDisabled	of_UnRegister
of_IsRegistered	

of_GetCount

Description Retrieves the number of registered objects (either zero or one).

Access Public

Syntax *tmgcontrolinstancename.of_GetCount ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)

Return value Integer. Returns 1 if an object is registered, 0 if no object is registered, and -1 if an error occurs.

Examples This example calls the of_GetCount function:

```
Integer li_return

IF IsValid(itmg_timer.inv_single) THEN
    li_return =  &
        itmg_timer.inv_single.of_GetCount()
END IF
sle_1.text = String(li_return)
```

of_GetInterval

Description Retrieves the current interval.

Access Public

Syntax *tmgcontrolinstancename.of_GetInterval ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)

Return value Real. Returns the current interval if the function succeeds and 0 if no interval has been defined.

Examples

This example calls the of_GetInterval function:

```
Real lr_interval

IF IsValid(itmg_timer.inv_single) THEN
    lr_interval = &
        itmg_timer.inv_single.of_GetInterval()
END IF
sle_1.text = "Interval is: " &
    + String(lr_interval)
```

of_GetRegistered**Description**

Retrieves information for the currently registered object.

Access

Public

Syntax

tmgcontrolinstancename.of_GetRegistered (object, event, interval, style)

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)
<i>object</i>	PowerObject into which the function places a reference to the registered object (passed by reference)
<i>event</i>	String into which the function places a reference to the registered event (passed by reference)
<i>interval</i>	Real into which the function places a reference to the interval (passed by reference)
<i>style</i>	Style into which the function places a reference to the notification style (passed by reference)

Return value

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

Examples

This example calls the of_GetRegistered function:

```
PowerObject lpo_object
String ls_event
Real lr_interval
Integer li_style

IF IsValid(itmg_timer.inv_single) THEN
    itmg_timer.inv_Single.of_GetRegistered &
        (lpo_object, ls_event, lr_interval, li_style)
END IF
...
```

of_IsAutoDisable

Description Reports whether the service is using the autodisable feature.

Access Public

Syntax *tmgcontrol.instancename.of_IsAutoDisable ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)

Return value Boolean. Returns TRUE if autodisable is enabled and FALSE if it is not.

Examples This example is from the pfc_Timer event:

```
...
If of_IsAutoDisable() Then
    of_SetDisabled(False)
End If
...
```

of_IsDisabled

Description Reports whether notification is disabled. If autodisable is enabled, n_cst_tmgsingle disables itself automatically while notifying the registered object.

Access Public

Syntax *tmgcontrol.instancename.of_IsDisabled ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)

Return value Boolean. Returns TRUE if notification is disabled and FALSE if it is not.

Examples This example is from the pfc_Timer event:

```
If of_IsDisabled() Then
    Return 0
End If
...
```

of_IsRegistered

Description Reports whether the specified object or object/event combination is registered.

Access Public

Syntax *tmgcontrol.instancename.of_IsRegistered (object { , event })*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)
<i>object</i>	PowerObject referencing the object to be tested
<i>event</i> (optional)	Event defined within <i>object</i>

Return value Boolean. Returns TRUE if *object* or *object* and *event* are registered with the service and FALSE if they are not.

Examples This example calls the of_IsRegistered function:

```
IF IsValid(itmg_timer.inv_single) THEN
    IF itmg_timer.inv_single.of_IsRegistered(parent)
THEN
    sle_1.text = "Window is registered"
ELSE
    sle_1.text = "Not registered"
END IF
```

of_Register

Description Registers an object instance (or object instance and event) with n_cst_tmgsingle.

Access Public

Syntax *tmgcontrol.instancename.of_Register (object, event, interval { , style })*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)
<i>object</i>	PowerObject referencing the instance to be notified when the timer fires
<i>event</i>	String specifying the event to be called. This must be a valid event for <i>object</i>

Argument	Description
<i>interval</i>	Real specifying the timer interval in seconds (.55 is the minimum and 65 is the maximum)
<i>style</i> (optional)	Integer or n_cst_tmgsingle constant specifying the notification style: <ul style="list-style-type: none"> • 0 or TRIGGEREVENT Trigger the event (default) • 1 or POSTEVENT Post the event

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

Usage Call this function to specify the object, event, and interval used by the timer.
If you specify an interval less than .55 seconds, the function sets the interval to .55; if you specify an interval greater than 65 seconds, the function sets the interval to 65.

Examples This example calls the of_Register function:

```
IF IsValid(itmg_timer.inv_single) THEN
    itmg_timer.inv_Single.of_Register &
    (parent, "ue_showtimer", 15)
END IF
```

of_SetAutoDisable

Description Specifies whether n_cst_tmgsingle disables notifications automatically during the notification process.

Access Public

Syntax *tmgcontrolinstancename.of_SetAutoDisable (boolean)*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)
<i>boolean</i>	Boolean specifying whether the service disables notification automatically (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_SetAutoDisable function:

```
itmg_timer.inv_single.of_SetAutoDisable(TRUE)
```

of_SetDisabled

Description Specifies whether n_cst_tmgsingle sends notifications to registered objects.

Access Public

Syntax *tmgcontrolinstancename.of_SetDisabled (boolean)*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)
<i>boolean</i>	Boolean specifying whether the service sends notifications to registered objects (TRUE) or not (FALSE)

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

Usage If autodisable is enabled, the pfc_Timer event calls this function to prevent simultaneous notifications.

Examples This example is from the pfc_Timer event:

```
If of_IsAutoDisable() Then
    of_SetDisabled(True)
End If
```

of_SetRequestor

Description Associates the requesting object with this instance of n_cst_tmgsingle.

Access Public

Syntax *tmgcontrolinstancename.of_SetRequestor (requestor)*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of n_cst_tmgsingle (the n_tmg default is inv_single)
<i>requestor</i>	N_tmg-based timing object instance 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 n_tmg of_SetSingle function:

```
...
inv_single = Create n_cst_tmgsingle
inv_single.of_SetRequestor(this)
...
```

of_UnRegister

Description Unregisters the current notification object.

Access Public

Syntax *tmgcontrol.instancename.of_UnRegister ()*

Argument	Description
<i>tmgcontrol</i>	Instance name of the n_tmg-based timing control
<i>instancename</i>	Instance name of <i>n_cst_tmgsingle</i> (the n_tmg default is <i>inv_single</i>)

Return value Integer. Returns 1 if the function succeeds, 0 if no object is registered, and -1 if an error occurs.

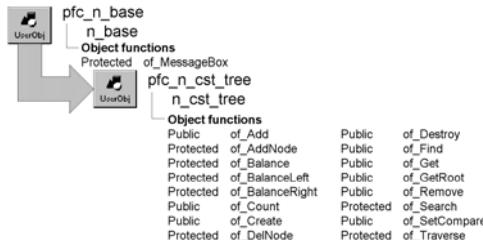
Examples This example calls the of_UnRegister function:

```
itmഗ_timer.inv_single.of_UnRegister()
```

n_cst_tree

Description Balanced binary tree object. A balanced binary tree provides an alternative to a sorted linked list. N_cst_tree automatically maintains a sorted (no duplicates) balanced binary tree, which is never more than one level out of balance.

Ancestry



Library PFCAPSrv.PBL
PFEAPSrv.PBL

Object relationships n_cst_treenode
n_cst_treenodecompare

Usage Use this object to handle list processing in a balanced binary tree.

To use n_cst_tree:

- 1 Create an instance of n_cst_tree. This example declares a window instance variable:

```
n_cst_tree inv_tree
```

- 2 (Optional) Specify a customized node comparison object. This example assumes an inv_customcompare instance variable of type n_cst_customcompare (n_cst_customcompare must be a descendant of n_cst_treenodecompare):

```
inv_customcompare = CREATE n_cst_mycompare

inv_tree.of_SetCompare(inv_customcompare)
```

- 3 Add nodes to the tree. To do this, first create a node, specify a key and data, then call the of_Add function:

```
n_cst_treenode lnv_node
Integer li_return

lnv_node = CREATE n_cst_treenode

lnv_node.of_SetKey(sle_1.text)
```

```
lnv_node.of_SetData(sle_1.text)
li_return = inv_tree.of_Add(lnv_node)
```

- 4 Access items in the tree as appropriate by calling the of_Find function.
This example displays the data in a MessageBox:

```
n_cst_treenode lnv_node, lnv_temp
Integer li_return
Any la_data

lnv_temp = CREATE n_cst_treenode

lnv_temp.of_SetKey(sle_2.text)
li_return = inv_tree.of_Find(lnv_node, lnv_temp)
IF li_return < 1 THEN
    MessageBox("Not Found", &
               "The specified node was not found")
ELSE
    lnv_node.of_GetData(la_data)
    MessageBox("Found", String(la_data) &
               + " was found.")
END IF
```

- 5 (Optional) Remove nodes from the tree by calling the of_Destroy function:

```
n_cst_treenode lnv_node, lnv_temp
Integer li_return
Long ll_oldtextcolor

lnv_temp = CREATE n_cst_treenode

lnv_temp.of_SetKey(sle_2.text)
li_return = inv_tree.of_Find(lnv_node, lnv_temp)
IF li_return < 1 THEN
    ll_oldtextcolor = sle_2.textcolor
    sle_2.textcolor = RGB(255, 0, 0)
    MessageBox("Not Found", &
               "The specified node was not found")
    sle_2.textcolor = ll_oldtextcolor
    Return
END IF
li_return = inv_tree.of_Destroy(lnv_node)
IF li_return = 1 THEN
    MessageBox("Success", &
               "Node removed successfully")
END IF
```

See also	n_cst_list n_cst_queue n_cst_stack n_cst_treenode n_cst_treenodecompare
----------	---

Instance variables

N_cst_tree includes instance variables:

Instance variable	Description	Data type	Access	Usage
inv_compare	Reference to comparison object	n_cst_treenodecompare	Protected	Set with of_SetCompare (default is n_cst_treenodecompare)
inv_removednode	Temporary node	n_cst_treenode	Protected	Internal
inv_root	Root node of tree	n_cst_treenode	Protected	Internal
inv_shiftnode	Temporary node	n_cst_treenode	Protected	Internal

Functions

N_cst_tree includes precoded object functions:

of_Add	of_Destroy
of_AddNode	of_Find
of_Balance	of_Get
of_BalanceLeft	of_GetRoot
of_BalanceRight	of_Remove
of_Count	of_Search
of_Create	of_SetCompare
of_DelNode	of_Traverse

of_Add

Description	Adds a new node to the tree.
Access	Public

Syntax	<i>instancename.of_Add (node)</i>
Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_tree</i>
<i>node</i>	<i>N_cst_treenode</i> instance to be added to the tree
Return value	Integer. Returns values as follows: <ul style="list-style-type: none">• 1 Success• 0 A node with this key already exists in the tree; <i>node</i> was not added• -1 An error occurred• -2 <i>Node</i> is invalid
Usage	Before calling this function, you need to create the node, add key information, and add data.
Examples	This example calls the <i>of_Add</i> function: <pre>n_cst_treenode lnv_node Integer li_return lnv_node = CREATE n_cst_treenode lnv_node.of_SetKey(sle_1.text) lnv_node.of_SetData(sle_1.text) li_return = inv_tree.of_Add(lnv_node)</pre>

of_AddNode

Description Finds the proper location for a new node and inserts it into the tree.

Access Protected

Syntax *instancename.of_AddNode (newnode, currentnode, height)*

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_tree</i>
<i>newnode</i>	<i>N_cst_treenode</i> instance containing the node to be added
<i>currentnode</i>	<i>N_cst_treenode</i> instance containing the current node (passed by reference)
<i>height</i>	Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (passed by reference)

Return value	Integer. Returns values as follows:
	<ul style="list-style-type: none"> • 1 Node was added successfully • 0 A node with a matching key already exists in the tree • -1 An error occurred
Usage	Internal.
Examples	This example calls the <code>of_AddNode</code> function:

```
...
Return of_AddNode &
(anv_currentnode, inv_root, lb_height)
```

of_Balance

Description Recursively balances the tree after a node deletion.

Access Protected

Syntax `instancename.of_Balance (node, height)`

Argument	Description
<code>instancename</code>	Instance name of <code>n_cst_tree</code>
<code>node</code>	<code>N_cst_treenode</code> instance referencing the top branch of the node to be balanced (passed by reference)
<code>height</code>	Boolean indicating whether the function changed the height of the tree (TRUE) or not (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_DelNode` function:

```
...
ELSE
    of_Balance(lnv_left, ab_height)
    inv_shiftnode.of_SetPrev(lnv_left)
...
...
```

of_BalanceLeft

Description Balances the left side of a node.

Access Protected

Syntax	<i>instancename.of_BalanceLeft (node, height)</i>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tree</td></tr> <tr> <td><i>node</i></td><td>N_cst_treenode instance referencing the node whose left side is to be balanced (passed by reference)</td></tr> <tr> <td><i>height</i></td><td>Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_tree	<i>node</i>	N_cst_treenode instance referencing the node whose left side is to be balanced (passed by reference)	<i>height</i>	Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (passed by reference)
Argument	Description								
<i>instancename</i>	Instance name of n_cst_tree								
<i>node</i>	N_cst_treenode instance referencing the node whose left side is to be balanced (passed by reference)								
<i>height</i>	Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (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_DelNode function:

```
...
IF ab_height THEN of_BalanceLeft &
    (anv_currentnode, ab_height)
...
```

of_BalanceRight

Description	Balances the right side of a node.								
Access	Protected								
Syntax	<i>instancename.of_BalanceRight (node, height)</i>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tree</td></tr> <tr> <td><i>node</i></td><td>N_cst_treenode instance referencing the node whose right side is to be balanced (passed by reference)</td></tr> <tr> <td><i>height</i></td><td>Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_tree	<i>node</i>	N_cst_treenode instance referencing the node whose right side is to be balanced (passed by reference)	<i>height</i>	Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (passed by reference)
Argument	Description								
<i>instancename</i>	Instance name of n_cst_tree								
<i>node</i>	N_cst_treenode instance referencing the node whose right side is to be balanced (passed by reference)								
<i>height</i>	Boolean indicating whether the function changed the height of the tree (TRUE) or not (FALSE) (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_DelNode function:

```
...
IF ab_height THEN of_BalanceRight &
    (anv_currentnode, ab_height)
...
```

of_Count

Description Retrieves a count of the number of nodes in the tree.

Access Public

Syntax *instancename.of_Count ()*

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree

Return value Long. Returns the number of nodes in the tree.

Examples This example calls the of_Count function:

```
...
    ll_count = inv_tree.of_Count()
    st_result.text = "Item count: " + String(ll_count)
```

of_Create

Description Creates a node, returning a reference to the new node.

Access Public

Syntax *instancename.of_Create (newnode)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree
<i>newnode</i>	N_cst_treenode instance into which the function places a reference to the newly created node (passed by reference)

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

Examples This example calls the of_Create function:

```
n_cst_treenode lnv_node
Integer li_return

inv_tree.of_Create(lnv_node)
lnv_node.of_SetKey(sle_1.text)
lnv_node.of_SetData(sle_1.text)
li_return = inv_tree.of_Add(lnv_node)
```

of_DeleteNode

Description Deletes the node whose key matches the key found in the passed argument, and if necessary rebalances the tree.

Access Protected

Syntax *instancename.of_DelNode (keynode, currentnode, height)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree
<i>keynode</i>	N_cst_treenode instance specifying the key to be deleted
<i>currentnode</i>	N_cst_treenode instance to be deleted from the tree (passed by reference)
<i>height</i>	Boolean (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_Remove function:

```
...
li_rc = of_DelNode(lnv_keynode, inv_root, lb_height)
inv_currentnode = inv_removednode
DESTROY lnv_keynode
Return li_rc
```

of_Destroy

Description Destroys the passed node, resetting references to surrounding nodes.

Access Public

Syntax *instancename.of_Destroy ({ node })*

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree
<i>node</i> (optional)	N_cst_treenode instance referencing the node to be destroyed. If you omit this argument, the function destroys all nodes in the tree

Return value Integer. Returns 1 if the function succeeds, -1 if *node* is not found, and -2 if *node* is invalid.

Examples This example calls the of_Destroy function:

```
n_cst_treenode lnv_node, lnv_temp
Integer li_return

lnv_temp = CREATE n_cst_treenode

lnv_temp.of_SetKey(sle_2.text)
li_return = inv_tree.of_Find(lnv_node, lnv_temp)
IF li_return < 1 THEN
```

```

        MessageBox( "Not Found" , &
        "The specified node was not found" )
        Return
    END IF
    li_return = inv_tree.of_Destroy(lnv_node)
    IF li_return = 1 THEN
        MessageBox( "Success" , &
        "Node removed successfully" )
    END IF

```

of_Find

Description Retrieves a reference to a node in the tree based on a specified key value.

Access Public

Syntax *instancename.of_Find (foundnode, nodewithkey)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree
<i>foundnode</i>	N_cst_treenode instance into which the function places the node if a match is found (passed by reference)
<i>nodewithkey</i>	N_cst_treenode instance containing the key value used to find the requested node

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

Usage Use *foundnode* to access the key and data for the node.

Examples This example calls the of_Find function:

```

n_cst_treenode lnv_node, lnv_temp
Integer li_return
Any la_data

lnv_temp = CREATE n_cst_treenode

lnv_temp.of_SetKey(sle_2.text)
li_return = inv_tree.of_Find(lnv_node, lnv_temp)
IF li_return < 1 THEN
    MessageBox( "Not Found" , &
    "The specified node was not found" )
ELSE
    lnv_node.of_GetData(la_data)
    MessageBox( "Found" , String(la_data) + " was found" )
END IF

```

of_Get

Description	Populates an array containing all nodes in the tree.
Access	Public
Syntax	<i>instancename.of_Get (allnodes)</i>
Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_tree</i>
<i>allnodes</i>	N_cst_treenode array into which the function places references to all nodes in the tree (passed by reference)
Return value	Long. Returns the number of elements in <i>allnodes</i> if the function succeeds and -1 if an error occurs.
Examples	This example calls the of_Get function:

```
n_cst_treenode lnv_nodes[ ]
Long ll_return, ll_count
Any la_data
String ls_data

lb_1.Reset()
ll_return = inv_tree.of_Get(lnv_nodes)
FOR ll_count = 1 to ll_return
    lnv_nodes[ll_count].of_GetData(la_data)
    ls_data = String(la_data)
    lb_1.AddItem(ls_data)
NEXT
```

of_GetRoot

Description	Retrieves a reference to the root node of a tree.
Access	Public
Syntax	<i>instancename.of_GetRoot (node)</i>
Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_tree</i>
<i>node</i>	N_cst_treenode instance into which the function places a reference to the root node of <i>instancename</i> (passed by reference)
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.

Examples

This example calls the of_GetRoot function:

```
n_cst_treenode lnv_node
Integer li_return
Any la_data

li_return = inv_tree.of_GetRoot(lnv_node)
IF li_return < 1 THEN
    MessageBox("Not Found",  &
              "Root node not found")
ELSE
    lnv_node.of_GetData(la_data)
    MessageBox("Root Node", "Root node is "  &
              + String(la_data))
END IF
```

of_Remove**Description**

Removes the specified node from the tree.

Access

Public

Syntax

instancename.**of_Remove** (*node*)

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree
<i>node</i>	N_cst_treenode instance referencing the node that contains the key and data to be removed (passed by reference)

Return value

Integer. Returns 1 if the function succeeds, -1 if an error occurs, and -2 if *node* is invalid.

Usage

In most cases it is better to call of_Destroy to remove and destroy the node.

Use with care

Due to the tree rotations that may occur while rebalancing the tree following the deletion, the data in the original node may have been moved. This function returns the node that was actually removed, but it may not be the node that was requested to be removed. If you require the removed node's contents after calling this function, make a copy of the node before calling the function.

Examples

This example is from the of_Destroy function:

```
...
li_rc = of_remove(anv_currentnode)
if li_rc = 1 then
    destroy anv_currentnode
end if
return li_rc
```

of_Search

Description

Searches the tree recursively until the specified node is found or the branch ends.

Access

Protected

Syntax

instancename.of_Search (foundnode, nodewithkey)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_tree</i>
<i>foundnode</i>	<i>N_cst_treenode</i> instance into which the function places the node if a match is found (passed by reference)
<i>nodewithkey</i>	<i>N_cst_treenode</i> instance containing the key value used to find the requested node

Return value

Integer. Returns 1 if the node was found and -1 if it is not found.

Usage

Internal.

Examples

This example is from the of_Find function:

```
...
anv_currentnode = inv_root
return of_search(anv_currentnode, anv_keynode)
```

of_SetCompare

Description

Establishes a comparison object other than *n_cst_treenodecompare*.

Access

Public

Syntax

instancename.of_SetCompare (comparisonobject)

Argument	Description
<i>instancename</i>	Instance name of <i>n_cst_tree</i>
<i>comparisonobject</i>	<i>N_cst_treenodecompare</i> or <i>n_cst_treenodecompare</i> descendant that overrides the of_Compare function

Return value	Integer. Returns 1 if the function succeeds, -1 if an error occurs, and -2 if the comparison object has already been set.
Usage	N_cst_tree uses the comparison object's of_Compare function to find and sort nodes. The default comparison object (n_cst_treenodecompare) can handle comparison for simple data types. But if your list's key items contain some other data type, you must create a customized n_cst_treenodecompare descendant and implement an of_Compare function to handle sort order.
Examples	This example calls the of_SetCompare function. It assumes an inv_customcompare instance variable:

```
inv_customcompare = CREATE n_cst_mycompare
inv_tree.of_SetCompare &
(inv_customcompare)
```

of_Traverse

Description	Walks the tree, creating a sorted array of nodes.
Access	Protected
Syntax	<i>instancename.of_Traverse (currentnode, foundnodes, count)</i>

Argument	Description
<i>instancename</i>	Instance name of n_cst_tree
<i>currentnode</i>	N_cst_treenode instance from which the function begins traversing the tree
<i>foundnodes</i>	N_cst_treenode array into which the function places found nodes in sorted order (passed by reference)
<i>count</i>	Long into which the function places a count of the elements in <i>foundnodes</i> (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_Get function:

```
Long ll_cnt = 0

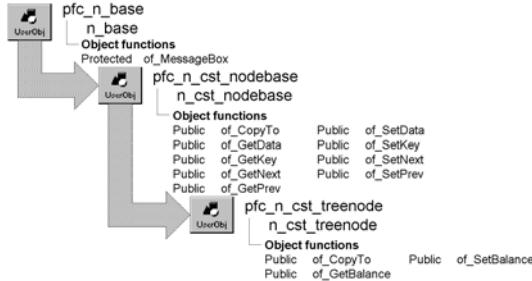
SetPointer(HourGlass!)
IF IsValid(inv_root) THEN
    of_Traverse(inv_root, anv_list, ll_cnt)
END IF
Return ll_cnt
```

n_cst_treenode

Description

Tree list instance object. Each element in a balanced binary tree is an instance of n_cst_treenode. N_cst_treenode contains balance information as well as functions to copy nodes and balance the tree.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

Use n_cst_treenode to create nodes for use with n_cst_tree.

To use n_cst_treenode:

- 1 Declare a variable for your list (an instance variable in this example):

```
n_cst_tree inv_tree
```

- 2 Declare variables for tree nodes (local variables in this example):

```
n_cst_treenode lInv_node
```

- 3 Create the node instance:

```
lInv_node = CREATE n_cst_treenode
```

You can also call the n_cst_tree of_Create function to create a tree node instance.

- 4 Access the node's key and data values as appropriate:

```
lInv_node.of_SetKey(sle_1.text)  
lInv_node.of_SetData(sle_1.text)
```

- 5 Call tree processing functions as appropriate:

```
li_return = inv_tree.of_Add(lInv_node)
```

See also

n_cst_linkedlistnode
n_cst_tree
n_cst_treenodecompare

Instance variables

N_cst_treenode includes instance variables:

Instance variable	Description	Data type	Access	Usage
ii_balance	Current tree balance	Integer	Protected	Set with of_SetBalance

Functions

N_cst_treenode includes precoded object functions:

of_CopyTo	of_SetBalance
of_GetBalance	

of_CopyTo

Description Copies the data and key of the passed node into the current node.

Access Public

Syntax *instancename.of_CopyTo (targetnode)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_treenode
<i>targetnode</i>	N_cst_treenode instance whose data and key are copied to <i>instancename</i> (passed by reference)

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

Usage If you extend n_cst_treenode, be sure to override this function to copy all added data elements as well as the ia_key and ia_data instance variables.

Examples

This example calls the of_CopyTo function:

```
n_cst_treenode lnv_node, lnv_temp, lnv_copy
Long ll_return
Any la_data

lnv_temp = CREATE n_cst_treenode

lnv_temp.of_SetKey(sle_2.text)
ll_return = inv_tree.of_Find(lnv_node, lnv_temp)
IF ll_return < 1 THEN
    MessageBox("Not Found", &
               "The specified node was not found")
    Return
END IF
lnv_copy = CREATE n_cst_treenode
ll_return = lnv_node.of_CopyTo(lnv_copy)
IF ll_return = 1 THEN
    lnv_copy.of_GetData(la_data)
    MessageBox("Success", &
               "Node copied successfully~r~n" &
               + "Data is: " + String(la_data))
END IF
```

of_GetBalance

Description

Retrieves the current balance of a tree list.

Access

Public

Syntax

instancename.**of_GetBalance** ()

Argument	Description
<i>instancename</i>	Instance name of n_cst_treenode

Return value

Integer. Returns values as follows:

- **1** Right branch is one level higher than the left
- **0** Tree is balanced
- **-1** Left branch is one level higher than the right

Usage

Internal.

Examples

This example calls the of_GetBalance function:

```
...
choose case anv_currentnode.of_GetBalance()
case 1
    anv_currentnode.of_SetBalance(0)
    ab_height = false
...
...
```

of_SetBalance**Description**

Sets the current balance in a tree list.

Access

Public

Syntax

instancename.of_SetBalance (*balancelevel*)

Argument	Description
<i>instancename</i>	Instance name of n_cst_treenode
<i>balancelevel</i>	<p>Integer specifying the balance level. Valid values are:</p> <ul style="list-style-type: none"> • 1 Right branch is one level higher than the left • 0 Tree is balanced • -1 Left branch is one level higher than the right

Return value

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

Examples

This example calls the of_SetBalance function:

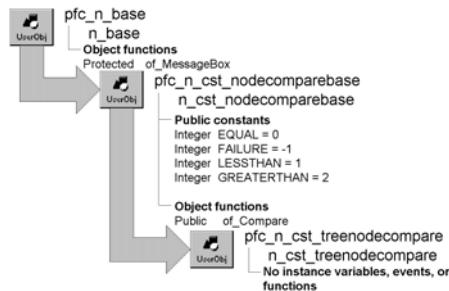
```
...
anv_newnode.of_SetBalance(0)
anv_currentnode = anv_newnode
...
...
```

n_cst_treenodecompare

Description

Tree node comparison object. N_cst_tree uses this object to compare values before placement in a tree list.

Ancestry



Library

PFCAPSRV.PBL
PFEAPSRV.PBL

Usage

Use this object for customized comparisons with tree lists.

To use a customized descendant of *n_cst_treenodecompare*:

- 1 Use the User Object painter to create a customized *n_cst_treenodecompare* descendant.
 - 2 In the customized *n_cst_treenodecompare* descendant, implement a Public *of_Compare* function to compare key values in the two passed nodes. This function should take two arguments of type *n_cst_treenode* (passed by value) and return Integer values as follows:
 - **LESSTHAN or 1** The key of the first node is less than the key of the second node
 - **EQUAL or 0** The key of the first node is equal to the key of the second node
 - **GREATERTHAN or 2** The key of the first node is greater than the key of the second node
- For an example of the processing you might use when overriding *of_Compare*, see *n_cst_linkedlistnodecompare* on page 1026.
- **FAILURE or -1** An error occurred

- 3 In the object that uses PFC tree processing, define an instance variable that uses your customized n_cst_treenodecompare object as the data type:

```
n_cst_mycompare inv_customcompare
```

- 4 Create an instance of the customized comparison object and call the n_cst_tree of_SetCompare function:

```
inv_customcompare = CREATE n_cst_mycompare
```

```
inv_tree.of_SetCompare(inv_customcompare)
```

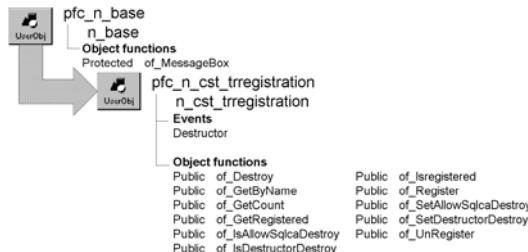
See also

[n_cst_linkedListnodecompare](#)
[n_cst_tree](#)
[n_cst_treenode](#)

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_SetAutoRollback function 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.

PFC 7 custom class extension

To further the extendibility of PFC, a new `n_cst_trregistrationattrib` custom class user object replaces the `os_transentry` object structure that is now obsolete.

See also	<code>n_cst_appmanager</code> <code>n_tr</code>
----------	--

Instance variables

`N_cst_trregistration` includes instance variables:

Instance variable	Description	Data type	Access	Usage
<code>ib_allowsqlcadestroy</code>	Indicates whether the service destroys SQLCA	Boolean	Protected	Set with <code>of_SetAllowSqlcaDestroy</code> (default is TRUE)
<code>ib_destructordestroy</code>	Indicates whether the service destroys registered transactions in the Destructor event	Boolean	Protected	Set with <code>of_SetDestructorDestroy</code> (default is TRUE)
<code>inv_trans[]</code>	Structure array that tracks Transaction objects and whether the array slot is used	<code>n_cst_trregistrationattrib</code>	Protected	Used to track Transaction objects
<code>ii_count</code>	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 <code>istr_trans</code> array

Events

`N_cst_trregistration` includes a precoded event:

Destructor

Destructor

Description Optionally destroys all registered Transaction objects.

Usage	Call the of_SetDestructorDestroy function to control whether n_cst_trregistration destroys registered Transaction objects in the Destructor event. 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_Destroy	of_IsRegistered
of_GetByName	of_Register
of_GetCount	of_SetAllowSqlcaDestroy
of_GetRegistered	of_SetDestructorDestroy
of_IsAllowSqlcaDestroy	of_UnRegister
of_IsDestructorDestroy	

of_Destroy

Description Unregisters and destroys all registered Transaction objects.

Access Public

Syntax *appnameinstancename.of_Destroy ()*

Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)

Return value Integer. Returns the number of destroyed Transaction objects if the function succeeds and -1 if an error occurs.

Usage To prevent this function from destroying SQLCA, call of_SetAllowSqlcaDestroy, passing FALSE.

Examples This example is from the Destructor event:

```
If of_IsDestructorDestroy() Then
    of_Destroy()
End If
```

of_GetByName

Description

Retrieves a reference to a registered Transaction object, given its name.

Access

Public

Syntax

`appnameinstancename.of_GetByName (transname, transobject)`

Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
<i>transname</i>	String specifying the name of the Transaction object to retrieve
<i>transobject</i>	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 `appnameinstancename.of_GetCount ()`

Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	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 `appnameinstancename.of_GetRegistered (transobj)`

Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
<i>transobj</i>	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_IsAllowSqlcaDestroy

Description

Reports whether the of_Destroy function destroys SQLCA.

Access

Public

Syntax

`appnameinstancename.of_IsAllowSqlcaDestroy ()`

Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)

Return value

Boolean. Returns TRUE if automatic SQLCA destroy is enabled and FALSE if it is not.

Examples

This example calls the of_IsAllowSqlcaDestroy function:

```
IF gnv_app.inv_trregistration.of_IsAllowSqlcaDestroy &
() THEN
    MessageBox( "TR Registration", &
    "Automatic SQLCA destroy enabled")
ELSE
    MessageBox( "TR Registration", &
    "Automatic SQLCA destroy disabled")
END IF
```

of_IsDestructorDestroy

Description

Reports whether the Destructor event destroys registered transactions.

Access

Public

Syntax

`appnameinstancename.of_IsDestructorDestroy ()`

Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)

Return value

Boolean. Returns TRUE if Destructor destroy is enabled and FALSE if it is not.

Examples

This example calls the of_IsDestructorDestroy function:

```
IF gnv_app.inv_trregistration.of_IsDestructorDestroy &
() THEN
    MessageBox( "Destructor", "Destroy enabled")
```

```

        ELSE
            MessageBox( "Destructor" , "Destroy disabled" )
    END IF

```

of_IsRegistered

Description Reports whether a Transaction object is registered with *n_cst_trregistration*.

Access Public

Syntax *appname.instancename.of_IsRegistered (nameortrans)*

Argument	Description
<i>appname</i>	Instance name of the <i>n_cst_appmanager</i> -based Application object (must be <i>gnv_app</i>)
<i>instancename</i>	Instance name of <i>n_cst_trregistration</i> (the <i>n_cst_appmanager</i> default is <i>inv_trregistration</i>)
<i>nameortrans</i>	<p>Either of the following:</p> <ul style="list-style-type: none"> • String containing the transaction name, as specified in the <i>n_tr of_SetName</i> function • <i>N_tr</i> 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
<i>appname</i>	Instance name of the <i>n_cst_appmanager</i> -based Application object (must be <i>gnv_app</i>)
<i>instancename</i>	Instance name of <i>n_cst_trregistration</i> (the <i>n_cst_appmanager</i> default is <i>inv_trregistration</i>)

Argument	Description
<i>transobject</i>	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_SetAllowSqlcaDestroy

Description	Specifies whether the of_Destroy function destroys SQLCA.
Access	Public
Syntax	<code>appnameinstancename.of_SetAllowSqlcaDestroy (boolean)</code>
Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
<i>boolean</i>	Boolean indicating whether the of_Destroy function destroys SQLCA (TRUE) or not (FALSE)
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Usage	The n_cst_trregistration default for this specification is TRUE, which enables automatic SQLCA destruction.
Examples	This example calls the of_SetAllowSqlcaDestroy function:

```

gnv_app.inv_trregistration.of_SetAllowSqlcaDestroy &
(FALSE)

```

of_SetDestructorDestroy

Description	Specifies whether the Destructor event destroys registered transactions.
Access	Public

Syntax	<code>appnameinstancename.of_SetDestructorDestroy (boolean)</code>
Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
<i>boolean</i>	Boolean indicating whether the Destructor event destroys registered transactions (TRUE) or not (FALSE)
Return value	Boolean. Returns TRUE if Destructor destroy is enabled and FALSE if it is not.
Usage	The n_cst_trregistration default for this specification is TRUE, which enables automatic transaction destruction.
Examples	This example calls the of_SetDestructorDestroy function:

```
gnv_app.inv_trregistration.of_SetDestructorDestroy &  
    ( FALSE )
```

of_UnRegister

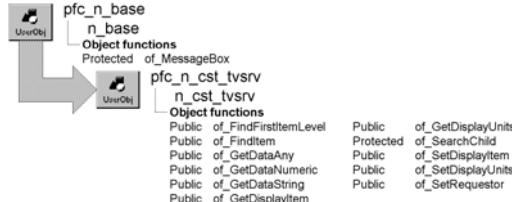
Description	Removes the specified Transaction object or all Transaction objects from the list of registered Transaction objects.
Access	Public
Syntax	<code>appnameinstancename.of_UnRegister ({ transobject })</code>
Argument	Description
<i>appname</i>	Instance name of the n_cst_appmanager-based Application object (must be gnv_app)
<i>instancename</i>	Instance name of n_cst_trregistration (the n_cst_appmanager default is inv_trregistration)
<i>transobject</i> (optional)	N_tr-based transaction variable to unregister. If you omit this argument, the function removes all registered Transaction objects
Return value	Integer. Returns the number of unregistered objects if the function succeeds and -1 if an error occurs.
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

N_cst_tvsrv includes 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 *tvcontrolinstancename.of_FindFirstItemLevel (level, handle)*

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tv-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv (the u_tv default is inv_base)
<i>level</i>	Integer specifying the TreeView level for which the first item is returned
<i>handle</i>	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.
1l_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

tvcontrolinstancename.of_FindItem (attribute, target {, beginhandle {, level {, respectcase {, fullcompare } } } })

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr (the u_tvs default is inv_base)
<i>attribute</i>	String specifying the TreeView attribute to search. Valid values are: <ul style="list-style-type: none"> • Label • Data
<i>target</i>	Any variable specifying the label or data that you are searching for
<i>beginhandle</i> (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
<i>level</i> (optional)	Integer specifying the level to search. Specify 0 to search all levels
<i>respectcase</i> (optional)	Boolean indicating whether the search is case sensitive (TRUE) or not (FALSE). The default is FALSE
<i>fullcompare</i> (optional)	Boolean indicating whether the specified attribute must match <i>target</i> exactly (TRUE) or the attribute can contain <i>target</i> . This argument applies only when <i>target</i> 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

tvcontrolinstancename.of_GetDataAny (handle)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsvr</i> (the u_tvs default is <i>inv_base</i>)
<i>handle</i>	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_currentttvitem
Any    la_data

ll_currentttvitem = tv_1.FindItem(currenttreeitem!, 0)
la_data = tv_1.inv_levelsource.of_GetDataAny  &
          (ll_currentttvitem)
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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr (the u_tvs default is inv_base)
<i>handle</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr (the u_tvs default is inv_base)
<i>handle</i>	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
<i>tvcontrol</i>	Instance name of the u_tvsv-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv (the u_tvsv 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
<i>tvcontrol</i>	Instance name of the u_tvsv-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv (the u_tvsv 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 <i>rows</i> , PFC displays Are you sure you want to delete these <i>rows</i> ?
Examples	This example calls the <code>of_GetDisplayUnits</code> 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	<code>tvcontrol.instancename.of_SearchChild (attribute, target, beginhandle, level respectcase, fullcompare)</code>

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr (the u_tvs default is <code>inv_base</code>)
<i>attribute</i>	String specifying the TreeView attribute to search. Valid values are: <ul style="list-style-type: none"> • Label • Data
<i>target</i>	Any variable specifying the label or data to search for
<i>beginhandle</i>	Long specifying the handle of the TreeView item from which to begin the search
<i>level</i>	Integer specifying the level to search
<i>respectcase</i>	Boolean indicating whether the search is case sensitive (TRUE) or not (FALSE)
<i>fullcompare</i>	Boolean indicating whether the specified attribute must match <i>target</i> exactly (TRUE) or the attribute can contain <i>target</i>

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
<code>tvcontrol</code>	Instance name of the u_tv-based TreeView control
<code>instancename</code>	Instance name of <i>n_cst_tvsrv</i> (the u_tv default is <i>inv_base</i>)
<code>itemname</code>	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
<code>tvcontrol</code>	Instance name of the u_tv-based TreeView control
<code>instancename</code>	Instance name of <i>n_cst_tvsrv</i> (the u_tv default is <i>inv_base</i>)
<code>units</code>	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
<i>tvcontrol</i>	Instance name of the <code>u_tvs</code> -based TreeView control
<i>instancename</i>	Instance name of <code>n_cst_tvsrv</code> (the <code>u_tvs</code> default is <code>inv_base</code>)
<i>requestor</i>	<code>U_tvs</code> -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
    ...
end if
```

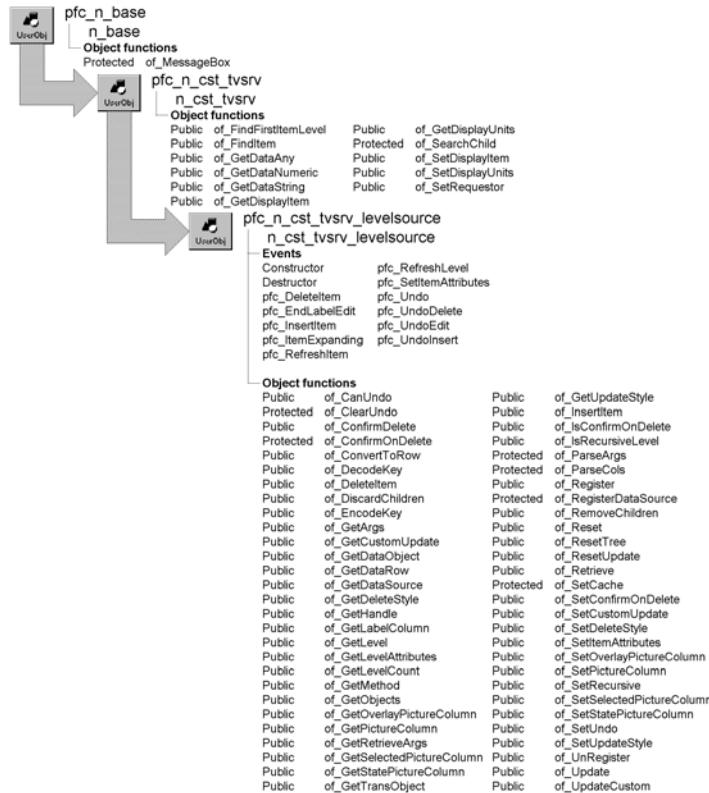
n_cst_tvsrv_levelsource

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_SetLevelSource function: this.of_SetLevelSource(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 Extend the pfc_Retrieve function, adding logic that handles retrieval arguments and calls the u_tvs of_Retrieve function: Any la_args[20] Integer li_level TreeViewItem ltvItem IF IsValid(this.inv_levelsource) THEN li_level = of_GetNextLevel(al_parent) this.inv_levelsource.of_GetArgs(al_parent, & li_level, la_args) END IF Return this.of_Retrieve & (al_parent, la_args, ads_data)
	5 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 n_cst_tvsrv_print u_tvs

Instance variables

N_cst_tvsrv_levelsouce includes instance variables:

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
CACHE_ID	Constant set to "level"	String	Public	Internal
CUSTOM	Constant set to 101	Integer	Public	Use with of_SetUpdateStyle
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
ib_isundoing	Indicates whether an undo is occurring	Boolean	Protected	Internal
ib_undo	Indicates whether undo is enabled	Boolean	Protected	Set with of_SetUndo
ii_customupdate	Custom update indicator	Integer	Protected	Set with of_SetCustomUpdate
ii_updatestyle	Indicates how PFC performs updates	Integer	Protected	Set with of_SetUpdateStyle
il_undodeletehandle	Deleted item	Long	Protected	Internal
il_undodeleteparenthandle	Parent of deleted item	Long	Protected	Internal
il_undodithandle	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_pfckey	Key value for cache	String	Protected	Internal
is_undoeeditlabeltext	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

Instance variable	Description	Data type	Access	Usage
UNDO_EDIT	Constant set to "Edit"	String	Public	Internal
UNDO_INSERT	Constant set to "Insert"	String	Public	Internal

Events

N_cst_tvsrv_levelsource includes pre-coded events:

Constructor	pfc_RefreshLevel
Destructor	pfc_SetItemAttributes
pfc_DeleteItem	pfc_Undo
pfc_EndLabelEdit	pfc_UndoDelete
pfc_InsertItem	pfc_UndoEdit
pfc_ItemExpanding	pfc_UndoInsert
pfc_RefreshItem	

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 destroyed.

pfc_DeleteItem

- Description Deletes the item from the TreeView and the row from the DataStore.
 Syntax `tvcontrol.instancename.EVENT pfc_DeleteItem (handle)`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)
<i>handle</i>	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 *tvcontrolinstancename.EVENT pfc_EndLabelEdit (handle, newlabel)*

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default for this value is inv_levelsource)
<i>handle</i>	Long specifying the handle of the modified TreeView item
<i>newlabel</i>	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 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_InsertItem

Description	Inserts a new item into the TreeView.
Syntax	<i>tvcontrol.instancename.EVENT pfc_InsertItem (parent, datastore, row, item, position, handle)</i>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsouce (the u_tvs default for this value is inv_levelsouce)
<i>parent</i>	Long specifying the handle of the TreeView item under which the new item is added. This argument is accessed through the <i>al_parent</i> argument
<i>datastore</i>	N_ds containing the data for the new item. This can be either the DataStore maintained by n_cst_tvsvr_levelsouce 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_tvsvr_levelsouce. This argument is accessed through the <i>ads_source</i> argument
<i>row</i>	Long specifying the row containing the item to be inserted. This argument is accessed through the <i>al_row</i> argument
<i>item</i>	TreeViewItem containing the item to be inserted. This argument is accessed through the <i>atvi_item</i> argument
<i>position</i>	String specifying the position under the current parent in which to insert the new item. Values are: <ul style="list-style-type: none">• First• Last• Sort• After This argument is accessed through the <i>as_position</i> argument
<i>handle</i>	Long containing the handle to the item before the item to be inserted. The function uses this argument if you specify After for <i>position</i> . This argument is accessed through the <i>al_handle</i> argument
Return value	Integer. Returns the handle of the new item if the function succeeds and -1 if an error occurs.
Usage	The u_tvs pfc_InsertItem event calls this event.
Examples	This example is from the u_tvs pfc_InsertItem event:

```

...
If isvalid(inv_levelsouce) then
    Return inv_levelsouce.event &
    pfc_InsertItem(al_parent, ads_source, al_row, &
```

```
        ltv_i_new, as_position, al_handle)
else
    Choose Case Lower(as_Position)
Case INSERT_LAST
    Return this.InsertItemLast(al_Parent,  &
        ltv_i_new)
...

```

pfc_ItemExpanding

Description Populates an item with its children.

Syntax **tvcontrol.instancename.EVENT pfc_ItemExpanding (handle)**

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default for this value is <i>inv_levelsource</i>)
<i>handle</i>	Long specifying the handle of the TreeView item being expanded. This argument is accessed through the <i>al_handle</i> argument

Usage The u_tvs ItemExpanding event calls this event.

Examples This example is from the u_tvs ItemExpanding event:

```
if IsValid(inv_levelsource) Then
    inv_levelsource.event pfc_ItemExpanding(handle)
end if
```

pfc_RefreshItem

Description Refreshes the specified TreeView item, resetting all properties to the defaults specified in the data source.

Syntax **tvcontrol.instancename.EVENT pfc_RefreshItem (handle)**

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default for this value is <i>inv_levelsource</i>)
<i>handle</i>	Long specifying the handle of the TreeView item to refresh. This argument is accessed through the <i>al_handle</i> argument

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

Usage The u_tvs pfc_RefreshItem event calls this event.

Examples

This example is from the u_tvs pfc_RefreshItem event:

```
If isvalid(inv_levelsource) then
    Return inv_levelsource.event &
        pfc_RefreshItem(al_handle)
End If
```

pfc_RefreshLevel**Description**

Refreshes the specified TreeView level, resetting all properties to the defaults specified in the data source.

Syntax

tvcontrol.instancename.EVENT pfc_RefreshLevel (/level)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default for this value is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level to refresh

Return value

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

Usage

The u_tvs pfc_RefreshLevel event calls this event.

Examples

This example is from the u_tvs pfc_RefreshLevel event:

```
If isvalid(inv_levelsource) then
    Return inv_levelsource.event pfc_RefreshLevel &
        (ai_level)
End IF
...
...
```

pfc_SetItemAttributes**Description**

Sets default properties for the TreeView item before insertion.

Syntax

tvcontrol.instancename.EVENT pfc_SetItemAttributes (data, row, tvitem)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default for this value is inv_levelsource)
<i>data</i>	N_ds containing the data for the item. This argument is accessed through the <i>ads_obj</i> argument

Argument	Description
<i>row</i>	Long specifying the row from which to populate the item. This argument is accessed through the <i>al_row</i> argument
<i>tvitem</i>	TreeViewItem to be inserted. This argument is accessed through the <i>atvi_item</i> argument (passed by reference)

Usage Optionally extend this event to change TreeView item properties before insertion.

pfc_Undo

Description Restores the last TreeView change.

Syntax *tvcontrol.instancename.EVENT pfc_Undo ()*

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default for this value is <i>inv_levelsource</i>)

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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	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 pre-coded functions:

of_CanUndo	of_GetUpdateStyle
of_ClearUndo	of_InsertItem
of_ConfirmDelete	of_IsConfirmonDelete
of_ConfirmOnDelete	of_IsRecursiveLevel
of_ConvertToRow	of_ParseArgs
of_DecodeKey	of_ParseCols
of_DeleteItem	of_Register
of_DiscardChildren	of_RegisterDataSource
of_EncodeKey	of_RemoveChildren
of_GetArgs	of_Reset
of_GetCustomUpdate	of_ResetTree
of_GetDataObject	of_ResetUpdate
of_GetDataRow	of_Retrieve
of_GetDataSource	of_SetCache
of_GetDeleteStyle	of_SetConfirmonDelete
of_GetHandle	of_SetCustomUpdate
of_GetLabelColumn	of_SetDeleteStyle
of_GetLevel	of_SetItemAttributes

of_GetLevelAttributes	of_SetOverlayPictureColumn
of_GetLevelCount	of_SetPictureColumn
ofGetMethod	of_SetRecursive
of_GetObjects	of_SetSelectedPictureColumn
of_GetOverlayPictureColumn	of_SetStatePictureColumn
of_GetPictureColumn	of_SetUndo
of_GetRetrieveArgs	of_SetUpdateStyle
of_GetSelectedPictureColumn	of_UnRegister
of_GetStatePictureColumn	of_Update
of_GetTransObject	of_UpdateCustom

of_CanUndo

Description

Reports whether the TreeView can undo the last edit, insertion, or deletion, optionally returning the operation type.

Access

Public

Syntax

tvcontrolinstancename.of_CanUndo ({ undotype })

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>undotype</i> (optional)	<p>String into which the function places the type of the last operation (passed by reference):</p> <ul style="list-style-type: none"> • 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
<code>tvcontrol</code>	Instance name of the u_tvs-based TreeView control
<code>instancename</code>	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
<code>tvcontrol</code>	Instance name of the u_tvs-based TreeView control
<code>instancename</code>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<code>amount</code>	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_ConfirmOnDelete

Description Reports whether a confirmation dialog box displays when the user deletes a row.

Access Protected

Syntax **tvcontrol.instancename.of_ConfirmOnDelete ()**

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Return value Boolean. Returns TRUE if confirm on delete is enabled and FALSE if it is disabled.

Usage Internal.

Examples This example is from the n_cst_tvsrv_levelsource of_DeleteItem function:

```
...
if of_ConfirmOnDelete() then
    li_confirm = of_ConfirmDelete(1)
...
...
```

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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>parent</i>	Long specifying the handle of the TreeView parent to which the row is added
<i>colvalues</i>	Any array containing the data to be used for the new row
<i>datastore</i>	N_ds instance into which the function places the new row (passed by reference)
<i>row</i>	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_DecodeKey

Description Retrieves the decoded key from the TreeView item's data attribute.

Access Public

Syntax `tvcontrol.instancename.of_DecodeKey (data)`

Argument	Description
<code>tvcontrol</code>	Instance name of the u_tvs-based TreeView control
<code>instancename</code>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<code>data</code>	String containing the ListView item's data attribute

Return value String. Returns the unique key if the function succeeds and ! if an error occurs.

Usage Internal.

Examples This example is from the pfc_UndoDelete event:

```

...
ls_undokey = of_DecodeKey(string(ltvi_undo.data))
...

```

of_DeleteItem

Description Removes either the specified item or all selected items from a TreeView level and the associated DataStore.

Access Public

Syntax	<code>tvcontrol.instancename.of_DeleteItem (handle)</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>handle</i>	Long specifying the handle of the item to be deleted
Return value	Integer. Returns 1 if the function succeeds, 0 if nothing was deleted, and -1 if an error occurs.
Usage	The pfc_DeleteItem and pfc_UndoInsert events call this function.
Examples	This example is from the pfc_DeleteItem event:

```
...
Return this.of_DeleteItem(al_handle)
...
```

of_DiscardChildren

Description	Removes from the TreeView all items below a specified item, discarding the rows from the associated DataStores.
Access	Public
Syntax	<code>tvcontrol.instancename.of_DiscardChildren (handle)</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>handle</i>	Long specifying the handle of the item whose children are to be deleted
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Usage	The pfc_RefreshItem event calls this function.
Examples	This example is from the pfc_RefreshItem event:

```
...
IF this.of_DiscardChildren(al_Handle) = -1 THEN
    Return -1
END IF
...
```

of_EncodeKey

Description Creates an encoded key from the TreeView item's data attribute. The service uses this key to access rows in the associated DataStore.

Access Public

Syntax `tvcontrolinstancename.of_EncodeKey (source, row)`

Argument	Description
<code>tvcontrol</code>	Instance name of the u_tvs-based TreeView control
<code>instancename</code>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<code>source</code>	N_ds containing the DataStore to populate the item
<code>row</code>	Long specifying the row containing the item

Return value String. Returns the unique key if the function succeeds and ! if an error occurs.

Usage Internal.

Examples This example is from the of_InsertItem function:

```

...
ls_key = this.of_EncodeKey  &
    (inv_attrib[li_Index].ids_obj, al_row)
...

```

of_GetArgs

Description Retrieves the retrieval arguments for a specified TreeView level.

Access Public

Syntax `tvcontrolinstancename.of_GetArgs (parent, level, arguments)`

Argument	Description
<code>tvcontrol</code>	Instance name of the u_tvs-based TreeView control
<code>instancename</code>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<code>parent</code>	Long containing the handle of the parent of the current TreeView item
<code>level</code>	Integer specifying the current TreeView level
<code>arguments</code>	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_GetCustomUpdate

Description

Retrieves the current custom update attribute.

Access

Public

Syntax

tvcontrolinstancename.of_GetCustomUpdate ()

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)

Return value

Integer. Returns the current custom update attribute.

Usage

The level source service architecture allows for multiple custom updates. Define an *of_UpdateCustom* function in the *n_cst_tvsrv_levelsource* 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.GetDataObject

Description

Retrieves the DataWindow object used as the level's data source.

Access

Public

Syntax

tvcontrolinstancename.of.GetDataObject (/level)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>handle</i>	Long specifying the handle for the TreeView item
<i>datasource</i>	N_ds specifying the DataStore that contains the row (passed by reference)
<i>row</i>	Long specifying the row in <i>datasource</i> that corresponds to <i>handle</i> (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

tvcontrolinstancename.of_GetDataSource (level, datastore)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>level</i>	Integer specifying the level for which a DataStore reference is returned
<i>datastore</i>	N_ds variable into which the function places a reference to the DataStore associated with <i>level</i> (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 ltv_i_item
n_ds lds_datastore

ll_current = tv_test.FindItem (CurrentTreeItem!, 0)
ll_return = tv_test.GetItem(ll_current, ltv_i_item)

ll_return = tv_test.inv_levelsource.of_GetDataSource &
            (ltv_i_item.level, lds_datastore)
IF ll_return = -1 THEN
    MessageBox("TreeView", "DataStore not found")
ELSE
    MessageBox("TreeView", &
                "Current level is: " &
                + String(ltv_i_item.level) &
```

```

+ " ~r~nDataWindow object is: " &
+ lds_datastore.DataObject)
END IF

```

of_GetDeleteStyle

Description Retrieves the current delete style.

Access Public

Syntax `tvcontrol.instancename.of_GetDeleteStyle ()`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)

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 DataStore buffer

Examples This example calls the of_GetDeleteStyle function:

```

Integer li_style

li_style =  &
    tv_1.inv_levelsource.of_GetDeleteStyle()
MessageBox("Delete Style", "Delete style is " &
    + String(li_style))

```

of_GetHandle

Description Retrieves the TreeView handle for a specified row in the DataStore.

Access Public

Syntax `tvcontrol.instancename.of_GetHandle (datasource, row{ , level })`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>datasource</i>	N_ds specifying the DataStore that contains the row (passed by reference)

Argument	Description
<i>row</i>	Long specifying the row in <i>datasource</i> for which the handle is returned
<i>level</i> (optional)	Integer specifying the level for which a DataStore reference is returned. Specify 0 to search the entire tree. If you don't specify this argument, the function uses the level for which <i>datasource</i> is registered

Return value Long. Returns the handle associated with *row* if the function succeeds, 0 if *row* is not found, and -1 if an error occurs.

Examples This example is from the pfc_RefreshLevel event:

```
...
For ll_row = 1 to ll_rowcount
    ll_handle = this.of_GetHandle(lds_source, &
        ll_row, ai_level)
    If ll_handle < 1 Then Continue
...

```

of_GetLabelColumn

Description Retrieves the name of the column used as the TreeView item label.

Access Public

Syntax *tvcontrolinstancename.of_GetLabelColumn (/level)*

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>level</i>	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	<code>tvcontrol.instancename.of_GetLevel (handle)</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>handle</i>	<p>Either of the following:</p> <ul style="list-style-type: none"> • 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 <i>handle</i> 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	<code>tvcontrol.instancename.of_GetLevelAttributes (level, tvattrib)</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)

Argument	Description
<i>level</i>	Integer specifying the TreeView level for which the function returns attributes
<i>tvattrib</i>	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.instancecname.of_GetLevelCount ()**

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancecname</i>	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	<i>tvcontrol.instancename.of_GetMethod (level)</i>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>tvcontrol</i></td><td>Instance name of the u_tvs-based TreeView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)</td></tr> <tr> <td><i>level</i></td><td>Integer specifying the TreeView level for which the function returns the population method</td></tr> </tbody> </table>	Argument	Description	<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control	<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)	<i>level</i>	Integer specifying the TreeView level for which the function returns the population method
Argument	Description								
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control								
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)								
<i>level</i>	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	<p>This example calls the of_GetMethod function:</p> <pre>String ls_method ls_method = tv_1.inv_levelsource.of_GetMethod(1) MessageBox("TreeView", "Method is: " + ls_method)</pre>								

of_GetObjects

Description	Retrieves a list of updatable data sources.								
Access	Public								
Syntax	<i>tvcontrol.instancename.of_GetObjects (objects)</i>								
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>tvcontrol</i></td><td>Instance name of the u_tvs-based TreeView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)</td></tr> <tr> <td><i>objects</i></td><td>PowerObject array into which the function places references to updatable objects (passed by reference)</td></tr> </tbody> </table>	Argument	Description	<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control	<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)	<i>objects</i>	PowerObject array into which the function places references to updatable objects (passed by reference)
Argument	Description								
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control								
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)								
<i>objects</i>	PowerObject array into which the function places references to updatable objects (passed by reference)								
Return value	Integer. Returns the number of elements in the <i>objects</i> 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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	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
<i>tvcontrol</i>	Instance name of the u_tvbs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvbs default is <i>inv_levelsource</i>)
<i>level</i>	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
<i>tvcontrol</i>	Instance name of the u_tvbs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvbs default is <i>inv_levelsource</i>)
<i>level</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level for which the function returns the population method
<i>trans</i>	N_tr into which the function places a reference to the Transaction object defined for <i>level</i> (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_GetUpdateStyle

Description Retrieves the update style for the current TreeView.

Access Public

Syntax	<code>tvcontrol.instancename.of_GetUpdateStyle()</code>						
	<table border="1"> <thead> <tr> <th>Argument</th><th>Description</th></tr> </thead> <tbody> <tr> <td><i>tvcontrol</i></td><td>Instance name of the u_tvs-based TreeView control</td></tr> <tr> <td><i>instancename</i></td><td>Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)</td></tr> </tbody> </table>	Argument	Description	<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control	<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
Argument	Description						
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control						
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)						
Return value	<p>Integer. Returns the update style:</p> <ul style="list-style-type: none"> • 1 or TOPDOWN • 2 or BOTTOMUP • 3 or TOPDOWN_BOTTOMUP • 4 or BOTTOMUP_TOPDOWN • 101 or CUSTOM 						
Examples	<p>This example calls the of_GetUpdateStyle function:</p> <pre>Integer li_count li_count = tv_1.inv_levelsource.of_GetUpdateStyle() MessageBox("Update Style", & + "Update style is " + String(li_count))</pre>						

of_InsertItem

Description	Adds a new item to the TreeView using a row from the data source.
Access	Public
Syntax	<code>tvcontrol.instancename.of_InsertItem(parent, datastore, row, tvitem, position, after)</code>

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>parent</i>	Long specifying the handle of the parent TreeView item
<i>datastore</i>	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)
<i>row</i>	Long specifying the row containing the new item
<i>tvitem</i>	TreeViewItem containing the item to be added to the TreeView

Argument	Description
<i>position</i>	String specifying the position under the current parent in which to insert the new item. Values are: <ul style="list-style-type: none">• First• Last• Sort• After
<i>after</i>	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 <i>position</i>
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, ltv_i_new, &
        as_position, al_handle)
End If
...

```

of_IsConfirmonDelete

Description Reports whether deletions are confirmed.

Access Public

Syntax **tvcontrol.instancename.of_IsConfirmonDelete ()**

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	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_levelsources.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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsources</i>)
<i>level</i>	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_levelsources.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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsources</i>)
<i>argument</i>	String containing the retrieval argument

Argument	Description
<i>level</i>	Integer specifying the TreeView level for which the arguments will be used
<i>arglevels</i>	Integer array into which the function places the TreeView levels used to create the retrieval argument (passed by reference)
<i>argcolumns</i>	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_ParseCols

Description Creates an array of columns from a passed string.

Access Protected

Syntax **tvcontrol.instancename.of_ParseCols (columns, operators, colarray)**

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsorce (the u_tvs default is inv_levelsorce)
<i>columns</i>	String containing the column definition string
<i>operators</i>	String into which the function places the relational operators (passed by reference)
<i>colarray</i>	String array into which the function places the columns to filter (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_Retrieve function:

```
...
Else

    of_ParseCols(inv_attrib[li_index].is_filtercols,  &
    ls_operators, ls_column)
...
...
```

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

Description

To display data retrieved via a DataWindow object

Establishes a TreeView level's data source by retrieving rows using the specified DataWindow object.

Access

Public

Syntax

tvcontrolinstancename.of_Register (level, labelcolumn, retrievespec, dwobject, trans, filterspec)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>labelcolumn</i>	String specifying the <i>dwobject</i> column that <i>u_tvs</i> uses for TreeView item labels
<i>retrievespec</i>	String containing the retrieval specification for <i>dwobject</i>

Argument	Description
<i>dwobject</i>	String specifying the DataWindow object used to retrieve data
<i>trans</i>	N_tr Transaction object used to retrieve rows for <i>dwobject</i> If you do not specify this argument, <i>dwobject</i> must have been saved with the data
<i>filterspec</i>	String containing the filter specification if required
Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none"> • 1 Success • -1 Argument validation error • -2 <i>Dwobject</i> had no key columns assigned • -3 A previous level was marked as recursive • -4 Column level datatype was not in the data source • -5 SetTransObject failed for the data source • -7 Cache registration error
Usage	<p>Call this function once for each TreeView level.</p> <p>The <i>retrieveargs</i> argument must be in the format:</p> <pre>:scope.level.column</pre> <p>where:</p> <ul style="list-style-type: none"> • <i>Scope</i> specifies either level or parent: <ul style="list-style-type: none"> • Level An absolute level number • Parent A number relative to the current level • <i>Level</i> 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
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**Description****To display data retrieved via a SQL statement**

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, retrievespec,
    trans, sql, filterspec )
```

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>labelcolumn</i>	String specifying the column that u_tvs uses for TreeView item labels
<i>retrievespec</i>	String containing the retrieval arguments if required
<i>trans</i>	N_tr Transaction object used to retrieve rows for <i>sql</i>
<i>sql</i>	String specifying the SQL SELECT statement used to create the TreeView level's data source
<i>filterspec</i>	String containing the filter specification if required

Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none"> • 1 Success • -1 Argument validation error • -2 A specified key column did not exist • -3 A previous level was marked as recursive • -4 Column level datatype was not in the data source • -5 SetTransObject failed for the data source • -7 Cache registration error
Usage	Call this function before calling other functions that customize TreeView display.
Examples	<p>This example calls the of_Register function:</p> <pre>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)</pre>

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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsources (the u_tvs default is inv_levelsources)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>labelcolumn</i>	String specifying the column that u_tvs uses for TreeView item labels
<i>retrievespec</i>	String containing the retrieval arguments if required
<i>dwcontrol</i>	DataWindow control containing data to use in the TreeView level
<i>filterspec</i>	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
- **-4** Column level datatype was not in the data source
- **-5** SetTransObject failed for the data source
- **-7** Cache registration error

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_levelsources.of_Register(1, "dept_name",  &
    "", dw_1, "")
this.inv_levelsources.of_SetPictureColumn(1, "1")
this.inv_levelsources.of_SetSelectedPictureColumn  &
    (1, "2")
this.inv_levelsources.of_Register(2, "emp_lname",  &
    ":parent.1.dept_id", "d_empbydept", SQLCA, "")
this.inv_levelsources.of_SetPictureColumn(2, "4")
this.inv_levelsources.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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>labelcolumn</i>	String specifying the column that u_tvs uses for TreeView item labels
<i>retrievespec</i>	String containing the retrieval arguments if required
<i>datastore</i>	DataStore containing data to use in the TreeView level
<i>filterspec</i>	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
- **-4** Column level datatype was not in the data source
- **-5** SetTransObject failed for the data source
- **-7** Cache registration error

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

Description

To display passed data

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
<i>tvcontrol</i>	Instance name of the u_tv-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tv default is <i>inv_levelsource</i>)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>labelcolumn</i>	String specifying the column that u_tv uses for TreeView item labels
<i>retrievespec</i>	String containing the retrieval arguments if required
<i>dwobject</i>	String specifying the DataWindow object whose columns correspond to <i>data</i>
<i>data</i>	PowerObject array containing the data to display in the TreeView level
<i>filterspec</i>	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
- **-4** Column level datatype was not in the data source
- **-5** SetTransObject failed for the data source
- **-7** Cache registration error

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 *dwoobject*.

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, dwoobject, file, filterspec)`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>labelcolumn</i>	String specifying the column that u_tvs uses for TreeView item labels
<i>retrievespec</i>	String containing the retrieval arguments if required
<i>dwoobject</i>	String specifying the DataWindow object whose columns correspond to the data in <i>file</i>
<i>file</i>	String specifying the file to be used as the TreeView level's data source
<i>filterspec</i>	String containing the filter specification if required

Return value	<p>Integer. Returns values as follows:</p> <ul style="list-style-type: none">• 1 Success• -1 Argument validation error• -2 <i>Dwobject</i> had no key columns assigned• -3 A previous level was marked as recursive• -4 Column level datatype was not in the data source• -5 SetTransObject failed for the data source• -7 Cache registration error
Usage	<p>Call this function before calling other functions that customize TreeView display.</p>

Elements must match

Errors will occur if the elements of the lines in *filename* do not exactly match the columns of *dwoject*.

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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsorce)
<i>level</i>	Integer specifying the TreeView level associated with this data source
<i>method</i>	String specifying the data source population method. Values are: Retrieve DataObjectData PowerObject DataWindowControl DataStoreControl ImportFile SQL
<i>dwobject</i>	String specifying the DataWindow object used by the data source
<i>trans</i>	N_tr specifying the Transaction object used by the data source
<i>sql</i>	String specifying the SQL statement used as the data source
<i>data</i>	PowerObject array containing data used as the data source
<i>dwcontrol</i>	DataWindow control whose associated DataWindow object and data is used as the data source
<i>datastore</i>	DataStore whose associated DataWindow object and data is used as the data source
<i>file</i>	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.

n_cst_tvsrv_levelsource

Access

Public

Syntax

tvcontrol.instancename.of_RemoveChildren (handle)

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>handle</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i> (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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>level</i> (optional)	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control

Argument	Description
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsouce (the u_tvs default is inv_levelsouce)
<i>level</i>	Integer specifying the TreeView level whose data source is retrieved
<i>arguments</i>	Twenty-element Any array containing retrieval arguments
<i>data</i>	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_levelsouce) then
    ll_rows = inv_levelsouce.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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsouce (the u_tvs default is inv_levelsouce)
<i>boolean</i>	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_levelsources displays a confirmation dialog box when users delete items.

Access Public

Syntax `tvcontrol.instancename.of_SetConfirmOnDelete (boolean)`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsources (the u_tvs default is inv_levelsources)
<i>boolean</i>	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_levelsources.of_SetConfirmOnDelete(TRUE)
this.event pfc_populateitem(0)
```

of_SetCustomUpdate

Description Specifies a custom update identifier.

Access Public

Syntax `tvcontrol.instancename.of_SetCustomUpdate (id)`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsources (the u_tvs default is inv_levelsources)
<i>id</i>	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:

```
tv_1.inv_levelsources.of_SetCustomUpdate(9)
```

of_SetDeleteStyle

Description Specifies whether deleted child rows are deleted or discarded.

Access Public

Syntax *tvcontrol.instancename.of_SetDeleteStyle (/level, style)*

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>level</i>	Integer specifying the TreeView level whose deletion style is specified
<i>style</i>	Integer or <i>n_cst_tvsrv_levelsource</i> constant specifying how to handle deleted children: <ul style="list-style-type: none"> • 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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>datastore</i>	N_ds containing the DataStore used to create TreeView item properties
<i>row</i>	Long specifying the row in <i>datastore</i> used to create TreeView item properties
<i>item</i>	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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levels (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level whose overlay picture is specified
<i>column</i>	String specifying one of the following: <ul style="list-style-type: none"> • 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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level whose picture is specified
<i>column</i>	String specifying one of the following: <ul style="list-style-type: none"> • 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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level
<i>boolean</i>	Boolean specifying whether <i>level</i> 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_levelsources.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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsources (the u_tvs default is inv_levelsources)
<i>level</i>	Integer specifying the TreeView level whose selected picture is specified
<i>column</i>	String specifying one of the following: <ul style="list-style-type: none"> • 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_levelsources.of_Register(2, "emp_lname", &
":parent.1.dept_id", "d_empbydept", SQLCA, "")
this.inv_levelsources.of_SetPictureColumn(2, "4")
this.inv_levelsources.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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>level</i>	Integer specifying the TreeView level whose state picture is specified
<i>column</i>	String specifying one of the following: <ul style="list-style-type: none">• 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_SetUndo

Description Specifies whether n_cst_tvsrv_levelsource enables undo processing.

Access Public

Syntax

`tvcontrolinstancename.of_SetUndo (boolean)`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>boolean</i>	Boolean specifying whether to enable (TRUE) or disable (FALSE) undo capabilities

Return value

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

Examples

This example calls the of_SetUndo function:

```
...
this.inv_levelsource.of_SetUndo(FALSE)
...
```

of_SetUpdateStyle**Description**

Specifies the update style.

Access

Public

Syntax

`tvcontrolinstancename.of_SetUpdateStyle (updatestyle)`

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>updatestyle</i>	<p>Integer or n_cst_tvsvr_levelsource constant specifying the update style:</p> <ul style="list-style-type: none"> • 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 levelservice service updates DataStores associated with TreeView levels.

If you specify a custom update style, you must also create and code an of_UpdateCustom function in the n_cst_tvsvr_levelsource extension-level object. This function then updates the DataStores in the desired order.

Examples

This example calls the of_SetUpdateStyle function:

```
tv_1.inv_levelsource.of_SetUpdateStyle  &
(tv_1.inv_levelsource.TOPDOWN)
```

of_UnRegister

Description

Removes a data source for the specified level.

Access

Public

Syntax

```
tvcontrol.instancename.of_UnRegister ( level )
```

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)
<i>level</i>	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

To 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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_levelsource</i> (the u_tvs default is <i>inv_levelsource</i>)

Argument	Description
<i>level</i>	Integer specifying the TreeView level whose DataStore is to be updated
<i>accept</i>	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database
<i>resetflag</i>	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

To update all TreeView levels

Description Saves all rows in all DataStores associated with the TreeView.

Access Public

Syntax *tvcontrolinstancename.of_Update (accept, resetflag)*

Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_levelsource (the u_tvs default is inv_levelsource)
<i>accept</i>	Boolean indicating whether the Update function performs an AcceptText before saving rows to the database
<i>resetflag</i>	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)
```

of_UpdateCustom

Description Empty function that you extend to implement custom update processing.

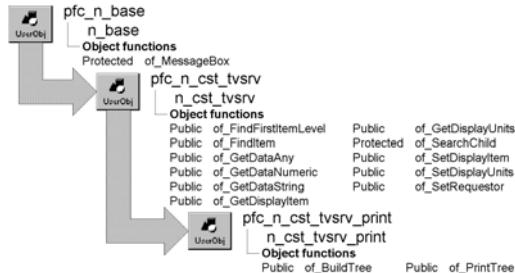
Access Public

Syntax	<code>tvcontrol.instancename.of_UpdateCustom (id)</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsrv_levelsource (the u_tvs default is inv_levelsource)
<i>id</i>	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_tvsrv_levelsource extension level object. This function checks the passed argument (if there are multiple custom update possibilities) and updates DataStores in the appropriate order.
Examples	This example is from the of_Update function: ... CASE CUSTOM li_rc = of_UpdateCustom(ii_customupdate) ...

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_levelsorce

Functions

N_cst_tvsrv_print includes pre-coded functions:

```

of_BuildTree
of_PrintTree
  
```

of_BuildTree

Description	Builds the data to be printed.
Access	Public
Syntax	<code>tvcontrol.instancename.of_BuildTree (handle, pictures, levelpicts)</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_print</i> (the u_tvs default is <i>inv_print</i>)
<i>handle</i>	Long containing the handle of the TreeView item from which to start building the data to print
<i>pictures</i>	String specifying the filenames for all pictures and selected pictures
<i>levelpicts</i>	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 <i>of_PrintTree</i> 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
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_PrintTree* 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	<code>tvcontrolinstancename.of_PrintTree ({ handle {, canceldialog {, pictname, selectedpictname } } })</code>
Argument	Description
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of n_cst_tvsvr_print (the u_tvs default is inv_print)
<i>handle</i> (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
<i>canceldialog</i> (optional)	Boolean specifying whether to display a Cancel dialog. The default is FALSE
<i>pictname</i> (optional)	String specifying the filename to display as the picture. The default is folder.bmp
<i>selectedpictname</i> (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

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
<i>tvcontrol</i>	Instance name of the u_tvs-based TreeView control
<i>instancename</i>	Instance name of <i>n_cst_tvsrv_print</i> (the u_tvs default is <i>inv_print</i>)
<i>handle</i> (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
<i>canceldialog</i> (optional)	Boolean specifying whether to display a Cancel dialog. The default is FALSE
<i>levelpicts</i> (optional)	Ten-element string array specifying pictures for items in up to ten levels
<i>levelselectedpicts</i> (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	<pre> graph TD pfc_n_base[pfc_n_base] --> n_base[n_base] n_base --> ObjectFunctions[Object functions] ObjectFunctions --> Protected[Protected of MessageBox] ObjectFunctions --> pfc_n_cst_winsrv[pfc_n_cst_winsrv] pfc_n_cst_winsrv --> n_cst_winsrv[n_cst_winsrv] n_cst_winsrv --> ObjectFunctions2[Object functions] ObjectFunctions2 --> Public[Public of Center] Public --> PublicOfSetRequestor[Public of SetRequestor] </pre>
Library	PFCWNSRV.PBL PFEWNSRV.PBL
Usage	<p>Use the functions in this user object to obtain basic information about a DataWindow object and its contents.</p> <p>To use this object:</p> <ol style="list-style-type: none"> 1 Enable the service using the w_master of_SetBase function: <pre>this.of_SetBase(TRUE)</pre> <ol style="list-style-type: none"> 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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv (the w_master default is inv_base)
<i>window</i>	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

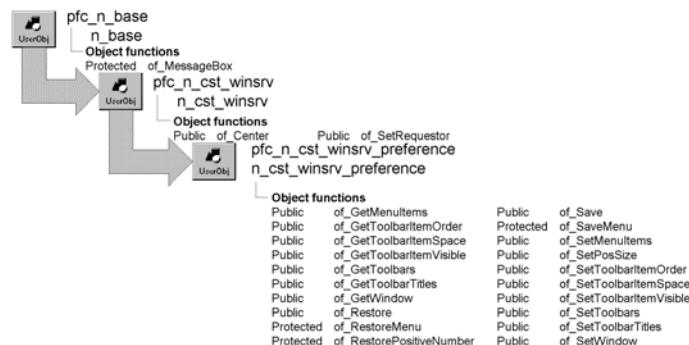
Description

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: <ol style="list-style-type: none">1 Create an instance of <code>n_cst_winsrv_preference</code> by calling the <code>w_master</code> of <code>_SetPreference</code> function.2 Specify the items to be restored by calling one or more of the following functions:<ul style="list-style-type: none">• The <code>of_SetMenuItems</code> function controls whether the service restores menu item properties• The <code>of_SetWindow</code> function controls whether the service restores window properties• The <code>of_SetToolbars</code> function controls whether the service restores toolbars• The <code>of_SetToolbarTitles</code> function controls whether the service restores toolbar titles• The <code>of_SetToolbarItemOrder</code> function controls whether the service restores the order of toolbar items• The <code>of_SetToolbarItemSpace</code> function controls whether the service restores the <code>ToolbarItemOrder</code> property• The <code>of_SetToolbarItemVisible</code> function controls whether the service restores the <code>ToolbarItemVisible</code> property3 Call the <code>of_Restore</code> function to reapply previous settings.4 Before closing the window, call the <code>of_Save</code> function to save window settings in the registry or an INI file.
See also	<code>n_cst_winsrv</code> <code>n_cst_resize</code> <code>n_cst_winsrv_sheetmanager</code> <code>n_cst_winsrv_statusbar</code> <code>w_master</code>

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
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 pre-coded object functions:

<code>of_GetMenuItems</code>	<code>of_Save</code>
<code>of_GetToolBarItemOrder</code>	<code>of_SaveMenu</code>
<code>of_GetToolBarSpaceItem</code>	<code>of_SetMenuItems</code>
<code>of_GetToolBarItemVisible</code>	<code>of_SetPosSize</code>
<code>of_GetToolbars</code>	<code>of_SetToolBarItemOrder</code>
<code>of_GetToolBarTitles</code>	<code>of_SetToolBarSpaceItem</code>
<code>of_GetWindow</code>	<code>of_SetToolBarItemVisible</code>
<code>of_Restore</code>	<code>of_SetToolbars</code>
<code>of_RestoreMenu</code>	<code>of_SetToolbarTitles</code>
<code>of_RestorePositiveNumber</code>	<code>of_SetWindow</code>

`of_GetMenuItems`

Description

Reports whether the service restores menu items.

Access

Public

Syntax

`instancename.of_GetMenuItems()`

Argument	Description
<code>instancename</code>	Instance name of <code>n_cst_winsrv_preference</code> (the <code>w_master</code> default is <code>inv_preference</code>)

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	<i>instancename</i> .of_GetToolbarItemOrder ()				
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
Argument	Description				
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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_Restore

There are also two undocumented versions of of_Restore, 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
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>registrykey</i>	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

Description

To restore settings from an INI file

Restores window size, position, toolbar settings, and menu item settings from an INI file.

Access

Public

Syntax

instancename.of_Restore (infile, section)

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>infile</i>	String specifying the INI file from which to restore settings
<i>section</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>useregistry</i>	Boolean specifying whether to use the registry (TRUE) or an INI file
<i>keyorinifile</i>	String specifying either the registry key or the INI file from which to restore settings

Argument	Description
<i>section</i>	String specifying the INI file section containing the settings
<i>menu</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>useregistry</i>	Boolean specifying whether to use the registry (TRUE) or an INI file (FALSE)
<i>keyorinifile</i>	String specifying either the registry key or the INI file from which to restore settings
<i>section</i>	String specifying the INI file section containing the settings
<i>key</i>	String specifying the registry or INI file key value
<i>value</i>	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

Description

To save settings in the registry

Saves window size, position, toolbar settings, and menu item settings in the registry.

Access

Public

Syntax

instancename.of_Save (registrykey)

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>registrykey</i>	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 (infile, section)

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>infile</i>	String specifying the INI file from which to restore settings
<i>section</i>	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	<i>instancename.of_SaveMenu (useregistry, keyorinifile, section, menu)</i>
<hr/>	
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>useregistry</i>	Boolean specifying whether to use the registry (TRUE) or an INI file
<i>keyorinifile</i>	String specifying either the registry key or the INI file in which to save settings
<i>section</i>	String specifying the INI file section to contain the settings
<i>menu</i>	Menu variable specifying the menu to be saved
<hr/>	
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_SetMenuItem

Description	Specifies whether the service restores menu items.
Access	Public
Syntax	<i>instancename.of_SetMenuItem (boolean)</i>

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>boolean</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>boolean</i>	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_SetBarItemOrder function:

```
Integer li_return

li_return = &

this.inv_preference.of_SetBarItemOrder(TRUE)
IF li_return = -1 THEN
    MessageBox("Preference", &
    "Error in of_SetBarItemOrder")
END IF
```

of_SetBarItemSpace

Description Specifies whether the service restores the ToolbarItemSpace property.

Access Public

Syntax *instancename.of_SetBarItemSpace (boolean)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>boolean</i>	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_SetBarItemSpace function:

```
Integer li_return

li_return = &

this.inv_preference.of_SetBarItemSpace(TRUE)
IF li_return = -1 THEN
    MessageBox("Preference", &
    "Error in of_SetBarItemSpace")
END IF
```

of_SetBarItemVisible

Description Specifies whether the service restores the ToolbarItemVisible property.

Access Public

Syntax *instancename.of_SetBarItemVisible (boolean)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>boolean</i>	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_SetBarItemVisible function:

```
Integer li_return

li_return = &

this.inv_preference.of_SetBarItemVisible(TRUE)
IF li_return = -1 THEN
    MessageBox("Preference", &
               "Error in of_SetBarItemVisible")
END IF
```

of_SetToolbars

Description Specifies whether the service restores toolbars.

Access Public

Syntax *instancename.of_SetToolbars (boolean)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>boolean</i>	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
<i>instancename</i>	Instance name of <i>n_cst_winsrv_preference</i> (the w_master default is <i>inv_preference</i>)
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_preference (the w_master default is inv_preference)
<i>boolean</i>	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:

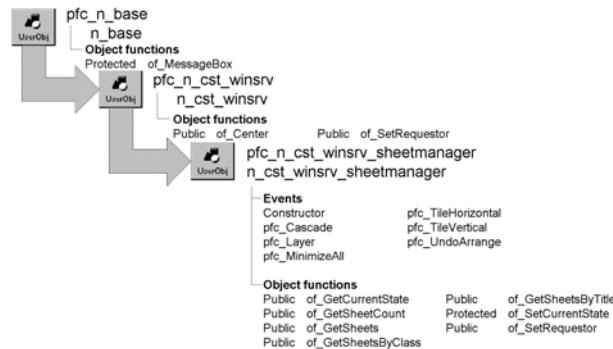
```
Integer li_return

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.

PFC 7 custom class extension

To further the extendibility of PFC, a new `n_cst_winsrv_sheetmanagerattrib` custom class user object replaces the `os_sheet` object structure that is now obsolete.

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
<code>ie_arrange</code>	Current arrange type	<code>ArrangeTypes</code>	Protected	Internal
<code>inv_sheet[]</code>	Array containing information on the previous state of active sheets	<code>n_cst_winsrv_sheetmanagerattrib</code>	Protected	Internal

Events

`N_cst_winsrv_sheetmanager` includes pre-coded events:

<code>Constructor</code>	<code>pfc_TileHorizontal</code>
<code>pfc_Cascade</code>	<code>pfc_TileVertical</code>
<code>pfc_Layer</code>	<code>pfc_UndoArrange</code>
<code>pfc_MinimizeAll</code>	

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
<i>instancename</i>	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
<i>instancename</i>	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	<i>instancename.of_GetSheets (sheets)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
<i>sheets</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
<i>sheets</i>	Array of windows into which the function places all sheets of the specified class (passed by reference)
<i>class</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
<i>sheets</i>	Array of windows into which the function places all sheets with the specified title (passed by reference)
<i>title</i>	String containing the title to be searched for
<i>partialmatch</i> (optional)	Boolean indicating whether the sheet title can contain <i>title</i> (TRUE) or must match <i>title</i> (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
<i>instancename</i>	Instance name of n_cst_winsrv_sheetmanager (the w_frame default is inv_sheetmanager)
<i>arrange</i>	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

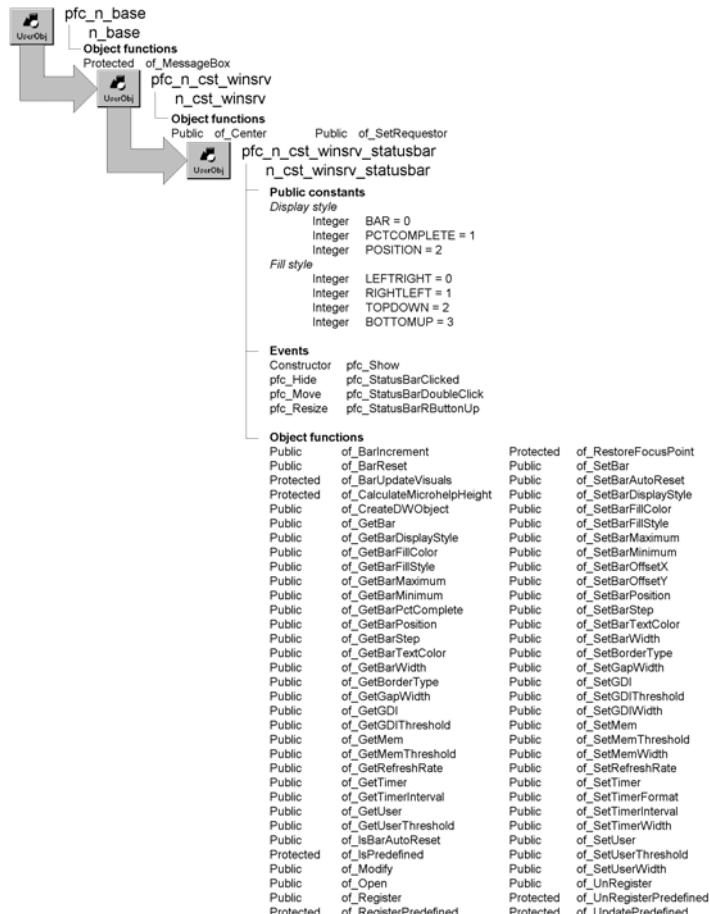
```

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.

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 functions as appropriate.

PFC 7 custom class extension

To further the extendibility of PFC, a new n_cst_winsrv_statusbarattrib custom class user object replaces the os_dwobjects object structure that is now obsolete.

See also

[n_cst_resize](#)
[n_cst_winsrv](#)
[n_cst_winsrv_preference](#)
[n_cst_winsrv_sheetmanager](#)
[w_frame](#)
[u_progressbar](#)
[w_statusbar](#)

Instance variables

N_cst_winsrv_statusbar includes 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)
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)

Instance variable	Description	Data type	Access	Usage
ii_barpcocomplete	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)
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

Instance variable	Description	Data type	Access	Usage
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
inv_dwobjects []	Status bar items	n_cst_winsrv_statusbar attrib	Protected	Internal
is_timerformat	Timer format	String	Protected	Set with of_SetTimerFormat (default is m-d-yy h:mm:ss)
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
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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar
<i>x</i>	Integer specifying the new x position of the associated frame window. From within the pfc_Move event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer specifying the new y position of the associated frame window. From within the pfc_Move event, access this value through the <i>ai_ypos</i> 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar
<i>sizetype</i>	Unsigned Long specifying the type of resize. Values are: <ul style="list-style-type: none">• 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 <i>aul_sizetype</i> argument
<i>newwidth</i>	Integer specifying the new window width, in PBUs
<i>newheight</i>	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	<i>instancename</i> .Event pfc_StatusBarClicked (<i>x</i> , <i>y</i> , <i>name</i>)

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar
<i>x</i>	Integer specifying the x position of the mouse click. From within the pfc_StatusBarClicked event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer specifying the y position of the mouse click. From within the pfc_StatusBarClicked event, access this value through the <i>ai_ypos</i> argument
<i>name</i>	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 <i>as_name</i> 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
<i>instancename</i>	Instance name of <i>n_cst_winsrv_statusbar</i>
<i>x</i>	Integer specifying the x position of the mouse click. From within the pfc_StatusBarDoubleClick event, access this value through the <i>ai_xpos</i> argument
<i>y</i>	Integer specifying the y position of the mouse click. From within the pfc_StatusBarDoubleClick event, access this value through the <i>ai_ypos</i> argument
<i>name</i>	String containing the name of the status bar element that was clicked (for example, this argument contains <i>pfc_mem</i> if the user clicks in the memory display). From within the pfc_StatusBarDoubleClick event, access this value through the <i>as_name</i> 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
<i>instancename</i>	Instance name of <i>n_cst_rtefind</i>
<i>x</i>	Integer specifying the x position of the mouse click. From within the pfc_StatusBarRButtonUp event, access this value through the <i>ai_xpos</i> argument

Argument	Description
<i>y</i>	Integer specifying the x position of the mouse click. From within the pfc_StatusBarRButtonUp event, access this value through the <i>ai_ypos</i> argument
<i>name</i>	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 <i>as_name</i> 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_BarLayout	of_RestoreFocusPoint
of_BarLayout	of_SetBar
of_BarLayout	of_SetBarAutoReset
of_CalculateMicrohelpHeight	of_SetBarDisplayStyle
of_CreateDWOObject	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>incrementvalue</i> (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_BarLayout` 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_BarLayout(1)
NEXT
lw_frame.inv_statusbar.of_SetBar(FALSE)

```

of_BarLayout**Description**

Clears the progress bar displayed in the status bar.

Access

Public

Syntax

`instancename.of_BarLayout()`

Argument	Description
<code>instancename</code>	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 of_BarLayout()
end if
...

```

of_BarLayout**Description**

Creates a string used to resize the progress bar and update the display text.

Access

Protected

Syntax

`instancename.of_BarLayout(completionvalue)`

Argument	Description
<code>instancename</code>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<code>completionvalue</code>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>createstatement</i>	String into which the function places syntax for the DataWindow Create statement (passed by reference)

Argument	Description
<i>width</i>	Integer into which the function places the width of the created object (passed by reference)
<i>height</i>	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
<i>instancename</i>	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   lInv_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 " &
    + lInv_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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of <i>n_cst_winsrv_statusbar</i> (the <i>w_frame</i> default is <i>inv_statusbar</i>)

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
<i>instancename</i>	Instance name of <i>n_cst_winsrv_statusbar</i> (the <i>w_frame</i> default is <i>inv_statusbar</i>)

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
<i>instancename</i>	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	<i>instancename.of_GetMem()</i>				
	<table><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
Argument	Description				
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	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
<i>instancename</i>	Instance name of <i>n_cst_winsrv_statusbar</i> (the <i>w_frame</i> default is <i>inv_statusbar</i>)

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
<i>instancename</i>	Instance name of <i>n_cst_winsrv_statusbar</i> (the <i>w_frame</i> default is <i>inv_statusbar</i>)

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
<i>instancename</i>	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	<i>instancename.of_IsBarAutoReset()</i>				
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
Argument	Description				
<i>instancename</i>	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	<i>instancename.of_IsPredefined(id)</i>						
	<table border="1"><thead><tr><th>Argument</th><th>Description</th></tr></thead><tbody><tr><td><i>instancename</i></td><td>Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)</td></tr><tr><td><i>id</i></td><td>String containing the ID to be tested</td></tr></tbody></table>	Argument	Description	<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)	<i>id</i>	String containing the ID to be tested
Argument	Description						
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)						
<i>id</i>	String containing the ID to be tested						
Return value	Boolean. Returns TRUE if <i>id</i> 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>id</i>	Either of the following: <ul style="list-style-type: none"> 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
<i>newvalue</i>	String containing either the new text or a bitmap filename. Do not specify this argument if <i>id</i> 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>id</i>	String specifying a unique identifier for the status bar object
<i>type</i>	String specifying whether the object displays text or a bitmap. Valid values are: <ul style="list-style-type: none">• text• bitmap
<i>value</i>	String specifying either the display text or bitmap name for the object
<i>width</i>	Integer specifying object width, in PBUs
<i>bordertype</i> (optional)	Integer specifying the border type. Values are: <ul style="list-style-type: none">• 0 None• 1 Shadow• 2 Rectangle• 3 Resize• 4 Line• 5 3D Lowered (default)• 6 3D Raised
<i>gapwidth</i> (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
<code>instancename</code>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<code>id</code>	String specifying one of the following: <ul style="list-style-type: none"> • Pfc_timer Time and date • Pfc_gdi GDI memory • Pfc_user User memory • Pfc_mem Free memory • Pfc_progress Progress bar
<code>width</code>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>control</i>	GraphicObject referencing the control to which focus is set. If <i>control</i> 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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	<i>instancename.of_SetBarDisplayStyle (displaystyle)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>displaystyle</i>	Integer or n_cst_winsrv_statusbar constant specifying the progress bar display style: <ul style="list-style-type: none">• 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>color</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar
<i>style</i>	Integer or n_cst_winsrv_statusbar constant specifying the fill style: <ul style="list-style-type: none"> 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>maximum</i>	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_BarLayout 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>maximum</i>	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_BarLayout 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>offset</i>	Integer specifying an <i>x</i> 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>offset</i>	Integer specifying a <i>y</i> 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	<i>instancename.of_SetBarPosition (position)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>position</i>	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 <i>position</i> is less than 0, this function sets the position to the minimum. If <i>position</i> 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_SetBarMaximummaximum 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_BarLayoutment(1)) NEXT

of_SetBarStep

Description	Specifies the step value that the progress bar uses during a default increment process.
Access	Public
Syntax	<i>instancename.of_SetBarStep (stepvalue)</i>
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar
<i>stepvalue</i>	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_BarLayoutment 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>color</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>width</i>	Integer specifying a new progress bar width, in PBUs

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

n_cst_winsrv_statusbar

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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>bordertype</i>	Integer specifying the border type. Values are: <ul style="list-style-type: none">• 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>width</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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	<i>instancename</i> .of_SetGDIThreshold (<i>threshold</i>)
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>threshold</i>	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:
	<pre>Integer li_threshold w_frame lw_frame lw_frame = gnv_app.of_GetFrame() li_threshold = & lw_frame.inv_statusbar.of_SetGDIThreshold() MessageBox("Status Bar", & "GDI memory threshold is " & + String(li_threshold))</pre>

of_SetGDIWidth

Description	Specifies the width for GDI display.
Access	Public
Syntax	<i>instancename</i> .of_SetGDIWidth (<i>width</i>)
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>width</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>threshold</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>width</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>refreshinterval</i>	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	<code>instancename.of_SetTimer (boolean)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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	<code>instancename.of_SetTimerFormat (format)</code>
Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>format</i>	String specifying the new date and time format. Sample formats include <i>mm/dd/yy h:mm:ss</i> , and <i>mmm dd, yyyy hh:mm:ss AM/PM</i>
Return value	Integer. Returns 1 if the function succeeds and -1 if an error occurs.
Examples	This example calls the of_SetTimerFormat function:
	<pre>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")</pre>

of_SetTimerInterval

Description Specifies a new refresh rate for the timer object.

Access Public

Syntax *instancename.of_SetTimerInterval (interval)*

Argument	Description
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>interval</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>width</i>	Integer specifying the width, in PBU's, 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>boolean</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>threshold</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>width</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>id</i>	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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>id</i>	<p>String specifying one of the following. Values are:</p> <ul style="list-style-type: none"> • 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
<i>instancename</i>	Instance name of n_cst_winsrv_statusbar (the w_frame default is inv_statusbar)
<i>id</i>	String specifying one of the following. Values are: <ul style="list-style-type: none">• Pfc_timer Timer• Pfc_gdi GDI memory• Pfc_user User memory• Pfc_mem Free memory• Pfc_progress Progress bar
<i>width</i>	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 12
 w_master 21
 w_sheet 58
application INI file
 of_GetAppIniFile 567
 of_SetAppIniFile 574
application manager 554
application services
 n_cst_apppreference 584
 n_cst_dwcache 651
 n_cst_error 936
 n_cst_security 1193
 n_cst_trregistration 1272

B

BeginDrag event 322
binary number service 1152
ButtonClicked event
 u_calculator 383
 u_calendar 408

C

calculator 380
calendar 403
CbnEditChange event
 u_cbx 120
 u_ddplb 127
CheckBox 115
Clicked event
 u_calendar 408
 u_dw 137

Close event
 w_frame 12
 w_master 21
CloseQuery event 21
code examples 7
color service 602
ColumnClick event 208
CommandButton 112
Commit 536
Connect 536
Constructor
 n_cst_mru 1125
Constructor event
 n_cst_appmanager 559
 n_cst_debug 636
 n_cst_dwsrv_report 828
 n_cst_filesrw32 992
 n_cst_inifile 1013
 n_cst_luw 1040
 n_cst_lvsrv_datasource 1062
 n_cst_platformwin32 1166
 n_cst_tmgtmultiple 1236
 n_cst_tvsrv_levelsources 1293
 n_cst_winsrv_sheetmanager 1373
 n_cst_winsrv_statusbar 1383
 u_calculator 383
 u_calendar 408
 u_progressbar 444
 u_st_splitbar 470
ContextInformation object 488
ContextKeyword object 488
ContextMenu event
 u_ddlb 120
 u_ddplb 127
 u_em 180
 u_lvs 208
 u_mle 238
 u_oc 244
 u_sle 292
conversion service 604

D

data type conversion service 604
DataStore 490
DataStore services
 n_cst_dssrv 647
 n_cst_dssrv_multitable 648
 n_cst_dssrv_printpreview 649
 n_cst_dssrv_report 650
DataWindow caching service
 main discussion 651
 of_SetDwCache 576
DataWindow control 132
DataWindow properties
 n_cst_dwsrv_property 811
 of_SetDWProperty 640
DataWindow services
 base object 665
 dropdown DataWindow search service 689
 filter service 700
 find service 711
 linkage service 727
 multitable update service 790
 n_cst_dwsrv_printpreview 802
 query mode service 815
 reporting 826
 required column 869
 resize service 877
 row manager 894
 row selection 904
 sort 920
 u_dw 132
date/time service 616
DBError event
 n_ds 493
 u_dw 138
debug service
 main discussion 634
 of_SetDebug 576
Destructor
 n_cst_mru 1125
Destructor event
 n_cst_appmanager 559
 n_cst_debug 636
 n_cst_dwcache 653
 n_cst_dwsrv_report 828
 n_cst_error 940

 n_cst_linkedlistbase 1018
 n_cst_luw 1062
 n_cst_tmgtmultiple 1236
 n_cst_trregistration 1273
 n_cst_tvsvr_levelsorce 1293
 n_ds 493
 n_tr 534
 u_base 366
 u_calculator 383
 u_calendar 408
 u_dw 138
 u_em 180
 u_lvs 208
 u_rte 270
 u_tvs 322
Disconnect 538
distributed computing
 pfc_ConnectionBegin 560
 pfc_ConnectionEnd 561
DoubleClick event 408
DragDrop event 322
dropdown calculator 380
dropdown calendar 403
dropdown DataWindows
 of_PopulateDDDW 684
 of_RefreshDDDWs 685
 search service 689
Dropdown event 138
dropdown service 641
DropDownListBox 118
DropDownPictureListBox 125
DynamicDescriptionArea object 489
DynamicStagingArea object 513

E

EditMask 179
EndLabelEdit event
 u_lvs 208
 u_tvs 322
Error object 514
error service
 main discussion 936
 of_SetError 577
examples 7

F

f_SetFilesrv 78
 f_SetPlatform 79
 file service
 f_SetFilesrv 78
 n_cst_filesrv 964
 FileExists event 270
 filter service 700
 find service 711
 frame menu 64

G

GetFocus event
 u_calculator 383
 u_calendar 409
 u_cb 112
 u_cbx 115
 u_ddlb 120
 u_ddplb 127
 u_dw 138
 u_em 181
 u_gr 188
 u_hpb 191
 u_hsb 194
 u_htb 197
 u_lb 200
 u_lvs 208
 u_mle 238
 u_oc 244
 u_p 251
 u_pb 254
 u_phl 257
 u_plb 260
 u_rb 264
 u_rte 270
 u_sle 292
 u_st 297
 u_sth 300
 u_tv 322
 u_vpb 353
 u_vsb 356
 u_vtb 359
 Graph 188
 GroupBox 187

H

Help
 accessing 6
 of_GetHelpFile 569
 of_SetHelpFile 578
 Hide event 13
 HorizontalProgressBar 191
 HorizontalScrollBar 194
 HorizontalTrackBar 197

I

Inet object 516
 INI file
 of_GetAppIniFile 567
 of_SetAppIniFile 574
 of_SetUserIniFile 581, 600, 1138
 INI file service 1012
 InternetResult object 516, 530
 ItemChanged event 138
 ItemError event 138
 ItemExpanding event 323
 ItemFocusChanged event 139

K

Key event
 u_calculator 383
 u_calendar 409
 KeyDown event
 u_lvs 208
 u_tvs 322

L

LButtonDown event 139
 LButtonUp event
 u_dw 139
 u_st_splitbar 470
 linkage service 727
 ListBox 200
 ListView 204

ListView service
base 1054
data source 1060
sort 1102

logical unit of work service 1038

logon
of_LogonDlg 573
pfc_Logon 562
pfc_PreLogonDlg 564

LoseFocus event

u_calculator 384
u_calendar 409

M

m_dw 60
m_edit 62
m_frame 64
m_lvs 65
m_master 68
m_oc 73
m_tvs 75

MailSession object 517

master menu 68
master/detail processing 727

menu service 1108

Message object 519

metaclass service 1114

MicroHelp

of_GetMicrohelp 570
of_SetMicrohelp 579
pfc_MicroHelp 13, 25, 58

most recently used object service 1121

MouseMove event 470

Move event 13, 21

MultiLineEdit 237

multitable update service

n_cst_dssrv_multitable 648
n_cst_dssrv_printpreview 649
n_cst_dwsrv_multitable 790

N

n_base 552

n_cn 484
n_cst_aboutattrib 82
n_cst_appmanager 554
n_cst_apppreference 584
n_cst_baseattrib 82
n_cst_calculatorattrib 83
n_cst_calendarattrib 83
n_cst_color 602
n_cst_columnattrib 84
n_cst_conversion 604
n_cst_datetime 616
n_cst_dberrorattrib 84
n_cst_debug
main discussion 634
n_cst_sqllsp 1205
n_cst_dirattrib 85
n_cst_dropdown 641
n_cst_dssrv 647
n_cst_dssrv_multitable 648
n_cst_dssrv_multitableattrib 86, 791
n_cst_dssrv_printpreview 649
n_cst_dssrv_report 650
n_cst_dwcache 651
n_cst_dwcacheattrib 86, 652
n_cst_dwobjectattrib 87
n_cst_dwpropertyattrib 87
n_cst_dwsrv 665
n_cst_dwsrv_dropdownsearch 689
n_cst_dwsrv_dropdownsearchattrib 88, 690
n_cst_dwsrv_filter 700
n_cst_dwsrv_find 711
n_cst_dwsrv_linkage 727
n_cst_dwsrv_multitable 790
n_cst_dwsrv_multitableattrib 88, 791
n_cst_dwsrv_printpreview 802
n_cst_dwsrv_property 811
n_cst_dwsrv_querymode 815
n_cst_dwsrv_querymodeattrib 89, 816
n_cst_dwsrv_report 826
n_cst_dwsrv_reqcolumn 869
n_cst_dwsrv_resize 877
n_cst_dwsrv_resizeattrib 90, 878
n_cst_dwsrv_rowmanager 894
n_cst_dwsrv_rowselection 904
n_cst_dwsrv_sort 920
n_cst_error 936

n_cst_errorattrib 91
n_cst_filesrv 964
n_cst_filesvrwin32 990
n_cst_filterattrib 92
n_cst_findattrib 93
n_cst_infoattrib 94
n_cst_inifile 1012
n_cst_itemattrib 94
n_cst_linkageattrib 95
n_cst_linkedlistbase 1017
n_cst_linkedlistnode 1025
n_cst_linkedlistnodecompare 1026
n_cst_list 1029
n_cst_logonattrib 95
n_cst_luw 1038
n_cst_lvssrv 1054
n_cst_lvssrv_datasource 1060
n_cst_lvssrv_sort 1102
n_cst_lvssrvattrib 96
n_cst_menu 1108
n_cst_metaclass 1114
n_cst_mru 1121
n_cst_mruattrib 97
n_cst_nodebase 1143
n_cst_nodecomparebase 1149
n_cst_numerical 1152
n_cst_platform 1158
n_cst_platformwin32 1165
n_cst_propertyattrib 97
n_cst_queue 1174
n_cst_resize 1177
n_cst_resizeattrib 98, 1178
n_cst_restorerowattrib 99
n_cst_returnattrib 99
n_cst_rtefind 1187
n_cst_security 1193
n_cst_selection 1197
n_cst_selectionattrib 100
n_cst_sortattrib 100
n_cst_splashattrib 101
n_cst_sql 1202
n_cst_sqllattrib 101
n_cst_sqllspy 1205
n_cst_stack 1211
n_cst_string 1214
n_cst_textstyleattrib 102
n_cst_tmgmultiple 1234
n_cst_tmgsingle 1244
n_cst_toolbarattrib 103
n_cst_tree 1253
n_cst_treenode 1266
n_cst_treenodecompare 1270
n_cst_trregistration 1272
n_cst_trregistrationattrib 103, 1273
n_cst_tvattrib 104
n_cst_tvssrv 1281
n_cst_tvssrv_levelsource 1290
n_cst_tvssrv_print 1347
n_cst_tvssrvattrib 105
n_cst_winsrv 1351
n_cst_winsrv_preference 1354
n_cst_winsrv_sheetmanager 1372
n_cst_winsrv_sheetmanagerattrib 106, 1372
n_cst_winsrv_statusbar 1379
n_cst_winsrv_statusbarattrib 106, 1380
n_cst_zoomattrib 107
n_cxinfo 488
n_cxk 488
n_dda 489
n_ds 490
n_dsa 513
n_err 514
n_inet 516
n_ir 516
n_ms 517
n_msg 519
n_oo 525
n_ostg 525
n_ostm 526
n_pl 527
n_srv 529
n_tmrg 530
 n_cst_tmgsingle 1234
 n_cst_tmgmultiple 1234
n_tr
 main discussion 532
 n_cst_trregistration 1272
n_trp 548
naming conventions
 instance variables 2
 objects 1

numerical service 1152

O

of_About 567

of_AcceptText

n_cst_dwsrv_linkage 740

n_cst_luw 1040

n_ds 499

u_base 369

u_dw 158

u_lvs 219

u_tab 307

u_tvs 334

w_master 38

of_Add

n_cst_linkedListbase 1018

n_cst_list 1031

n_cst_tree 1255

of_AddColumn 692

of_AddCompute 829

of_AddHead

n_cst_linkedListbase 1019

n_cst_list 1031

of_AddItem

n_cst_mru 1127

u_lvs 220

u_tvs 335

of_AddLine 832

of_AddMenubarLine 1128

of_AddNode 1256

of_AddPicture 835

of_AddTail

n_cst_linkedListbase 1019

n_cst_list 1032

of_AddText 837

of_AddToUpdate 792

of_ArrayToString 1215

of_Assemble 1202

of_AssemblePath 966

of_Balance 1257

of_BalanceLeft 1257

of_BalanceRight 1258

of_BarIncrement 1388

of_BarReset 1389

of_BarUpdateVisuals 1389

of_Begin 535

of_BeginTran 1041

of_Binary 1153

of_BitwiseAnd 1153

of_BitwiseNot 1154

of_BitwiseOr 1155

of_BitwiseXor 1156

of_Boolean 605

of_BuildColumnNames 715

of_BuildComparison 667

of_BuildExpression 668

of_BuildFilterAttrib 702

of_BuildFindExpression 716

of_BuildSortAttrib 924

of_BuildTextModify 840

of_BuildTree 1348

of_Button 606

of_ButtonClicked 385

of_ButtonUp 911

of_CalculateFileAttributes 966

of_CalculateMicrohelpHeight 1390

of_CanUndo

n_cst_lvsrv_datasource 1069

n_cst_tvsrv_levelsources 1302

u_lvs 220

u_tvs 336

of_Center 1352

of_ChangeDirectory

n_cst_filesrv 967

n_cst_filesrwin32 993

of_CheckRequired

n_cst_dwsrv_linkage 740

n_ds 499

u_dw 158

of_ClearLog 636

of_ClearUndo

n_cst_lvsrv_datasource 1070

n_cst_tvsrv_levelsources 1303

of_Commit 536

of_Compare 1150

of_CompareAttrib

n_cst_dwsrv_find 716

n_cst_rtefind 1190

of_ConfirmDelete
 n_cst_dwsrv_rowmanager 897
 n_cst_lsrvr_datasource 1070
 n_cst_tvsrv_levelsouce 1303
 of_ConfirmOnDelete 741
 n_cst_lsrvr_datasource 1071
 n_cst_tvsrv_levelsouce 1304
 of_ConfirmOnRowChange 742
 of_Connect 536
 of_ConvertFileDialogTimeToPB
 n_cst_filesvrwin32 993
 of_ConvertPBDATimeToFile
 n_cst_filesvrwin32 994
 of_ConvertToRow 1072, 1304
 of_CopyTo
 n_cst_nodebase 1144
 n_cst_treenode 1267
 n_msg 520
 n_tr 537
 of_Count
 n_cst_linkedListbase 1020
 n_cst_tree 1259
 of_CountOccurrences 1216
 of_Create 1259
 of_CreateComposite 842
 of_CreateDirectory
 n_cst_filesvr 968
 n_cst_filesvrwin32 995
 of_CreateDWOObject 1390
 of_CreateLogText 941
 of_CreateNotifyText 941
 of_CreateUpdateSequence 742
 of_Date 606
 of_DayOfWeek 617
 of_Days 618
 of_DBError 1042
 of.Decimal 1156
 of_DecodeKey
 n_cst_lsrvr_datasource 1072
 n_cst_tvsrv_levelsouce 1305
 of_Delete 1013
 of_DeleteAll 897
 of_DeleteDetailRows 743
 of_DeleteItem 693
 n_cst_lsrvr_datasource 1073
 n_cst_tvsrv_levelsouce 1305
 of_DeleteRow 898
 of_DeleteRows 744
 of_DeleteSelected 898
 of_DelNode 1259
 of_DelTree 968
 of_Describe 669
 of_Destroy 1274
 n_cst_linkedListbase 1020
 n_cst_tree 1260
 of_DirAttribToDS 969
 of_DirectoryExists
 n_cst_filesvr 969
 n_cst_filesvrwin32 996
 of_DirList
 n_cst_filesvr 970
 n_cst_filesvrwin32 996
 of_DiscardChildren 1306
 of_Disconnect 538
 of_DistinctValues 538
 of_DrawMonth 412
 of_DropDown
 u_calculator 385
 u_calendar 412
 of_DSToDirAttrib 970
 of_DWArguments 671
 of_DWItemStatus 607
 of_EncodeKey
 n_cst_lsrvr_datasource 1073
 n_cst_tvsrv_levelsouce 1307
 of_End 539
 of_EndTran 1042
 of_EndTranError 1043
 of_Execute 539
 of_FileCopy 971
 of_FileRead 972
 of_FileRename
 n_cst_filesvr 974
 n_cst_filesvrwin32 997
 of_FileWrite 974
 of_Filter 703, 1129
 of_FilterDetails 745
 of_Find 717
 n_cst_list 1032
 n_cst_tree 1261
 of_FindFirstItemLevel 1282
 of_FindItem 1283

of_FindMatchingEvent 1115
of_FindMatchingVariable 1116
of_FindMRUItem 1129
of_FindNext 1055
of_FindRoot 745
of_FindSelected 1056
of_FindStartAndEndRows 718
of_FindWindow
 n_cst_platform 1159
 n_cst_platformwin32 1167
of_FirstDayOfMonth 618
of_FirstPage 803
of_FormatData 1074
of_GCommonDenominator 1236
of_Get 1175
 n_cst_linkedListbase 1021
 n_cst_tree 1262
of_GetAllToolbarIndex 1109
of_GetAltFilename
 n_cst_filesrv 976
 n_cst_filesrwin32 998
of_GetAlwaysOnTop
 n_cst_debug 637
 n_cst_sqlspy 1206
of_GetAncestorClasses 1117
of_GetAppend 500
of_GetAppIniFile 567
of_GetAppKey 568
of_GetArgs 1307
of_GetArguments 746
of_GetAttributes
 n_cst_lvsrv_datasource 1075
of_GetAutoRollback 540
of_GetBackColor 445
of_GetBalance 1268
of_GetBar 1392
of_GetBarAutoClear 1405
of_GetBarColor 471
of_GetBarDisplayStyle 1393
of_GetBarFillColor 1393
of_GetBarFillStyle 1394
of_GetBarMaximum 1395
of_GetBarMinimum 1395
of_GetBarMoveColor 471
of_GetBarPctComplete 1396
of_GetBarPosition 1396
of_GetBarStep 1397
of_GetBarTextColor 1398
of_GetBarWidth 1398
of_GetBeep 942
of_GetBit 1157
of_GetBorderType 1399
of_GetByName 1275
of_GetCloseStatus 38
of_GetColumn 693
of_GetColumnDisplayName 673
of_GetColumnDisplayStyle 673
of_GetColumnEditStatus 719
of_GetColumnHeader 924
of_GetColumnInfo
 n_cst_lvsrv_datasource 1075
 u_lvs 221
of_GetColumnLabel 1078, 1312
of_GetColumnNameSource 674
of_GetComputerName
 n_cst_platform 1160
 n_cst_platformwin32 1167
of_GetConfirmonDelete 899
of_GetContinuousPages 277
of_GetCopyright 568
of_GetCount
 n_cst_dwcache 653
 n_cst_tmgsingle 1237, 1246
 n_cst_trregistration 1276
of_GetCreationDate 976
of_GetCreationDateTime
 n_cst_filesrv 977
 n_cst_filesrwin32 999
of_GetCreationTime 977
of_GetCurrentDirectory
 n_cst_filesrv 978
 n_cst_filesrwin32 1000
of_GetCurrentState 1375
of_GetCurrentText 445
of_GetCustomUpdate
 n_cst_dwsrv_linkage 746
 n_cst_tvsrv_levelsouce 1308
of_GetData 1145
of_GetDataAny 1284
of_GetDataNumeric 1284

of_GetDataObject
 n_cst_lvsrv_datasource 1076
 n_cst_tvsrv_levelwsource 1308
 of_GetDataRow 1076, 1309
 of_GetDataSource 1077
 of_GetDataStore 1310
 of_GetDataString 1285
 of_GetDefaultBackColor 843
 of_GetDefaultBorder 844
 of_GetDefaultCharset 844
 of_GetDefaultColor 845
 of_GetDefaultFontFace 846
 of_GetDefaultFontSize 846
 of_GetDefaultHeaderSuffix 674
 of_GetDeleteStyle
 n_cst_dwsrv_linkage 747
 n_cst_tvsrv_levelsources 1311
 of_GetDetails 747
 of_GetDiskSpace
 n_cst_filesv 978
 n_cst_filesvwin32 1000
 of_GetDiskSpaceEx
 n_cst_filesv 978
 n_cst_filesvwin32 1000
 of_GetDisplayItem
 n_cst_dwsrv 675
 n_cst_lvsrv 1057
 n_cst_tvsrv 1286
 of_GetDisplayStyle 446
 of_GetDisplayUnits
 n_cst_dwsrv 675
 n_cst_lvsrv 1057
 n_cst_tvsrv 1286
 of_GetDoubleParm 520
 of_GetDriveType
 n_cst_filesv 979
 n_cst_filesvwin32 1001
 of_GetDWTypr 39
 of_GetEnabled
 n_cst_dwsrv_printpreview 804
 n_cst_dwsrv_querymode 818
 of_GetExclude
 n_cst_dwsrv_filter 703
 n_cst_dwsrv_sort 925
 n_cst_lvsrv_sort 1104
 of_GetExtremePoint 471
 of_GetFileAttributes
 n_cst_filesv 979
 n_cst_filesvwin32 1002
 of_GetFileName 278
 of_GetFileSize
 n_cst_filesv 980
 n_cst_filesvwin32 1003
 of_GetFillColor 446
 of_GetFillStyle 447
 of_GetFilter 704
 of_GetFrame 569
 of_GetFreeMemory
 n_cst_platform 1160
 n_cst_platformwin32 1168
 of_GetGapWidth 1400
 of_GetGDI 1400
 of_GetGDIThreshold 1401
 of_GetHandle 1311
 of_GetHeaderName 676
 of_GetHeight 677
 of_GetHelpFile 569
 of_GetHoliday 413
 of_GetHolidayColor 413
 of_GetHorizontalPointer 472
 of_GetIgnoreFileExists 278
 of_GetInfo
 n_cst_dwsrv 677
 n_cst_dwsrv_dropdownsearch 694
 n_cst_dwsrv_filter 705, 720
 n_cst_dwsrv_linkage 748
 n_cst_dwsrv_multitable 793
 n_cst_dwsrv_printpreview 804
 n_cst_dwsrv_querymode 818
 n_cst_dwsrv_report 846
 n_cst_dwsrv_reqcolumn 870
 n_cst_dwsrv_resize 881
 n_cst_dwsrv_rowmanager 900
 n_cst_dwsrv_rowselection 911
 n_cst_dwsrv_sort 925
 u_base 370
 u_calculator 386
 u_calendar 414
 u_lvs 222
 u_progressbar 447
 u_st_splitbar 473

Index

u_tab 308
u_tvs 336
w_master 40
of_GetInterval
 n_cst_tmgtmgmultiple 1237
 n_cst_tmgsingle 1246
of_GetItem 678
of_GetItemAny 679
of_GetItemIndex 1078
of_GetKey 1145
of_GetKeyboard 914
of_GetKeys 1014
of_GetKeyValue 1216
of_GetLastAccessDate
 n_cst_filesrv 980
 n_cst_filesrwin32 1004
of_GetLastWriteDate 980
of_GetLastWriteDateTime
 n_cst_filesrv 981
 n_cst_filesrwin32 1005
of_GetLastWriteTime 981
of_GetLevel 1313
of_GetLevelAttributes 1313
of_GetLevelCount 1314
of_GetLineEnding 1015
of_GetLogFile 942
of_GetLogFileStyle 943
of_GetLogo 570
of_GetLogSeverity 943
of_GetLongFilename
 n_cst_filesrv 982
 n_cst_filesrwin32 1006
of_GetMarkedDay 415
of_GetMarkedDayColor 415
of_GetMaster 749
of_GetMaximum 448
of_GetMDIFrame 1109
of_GetMem 1401
of_GetMemThreshold 1402
of_GetMenuItem 1357
of_GetMenuReference 1110
of_GetMessageText 448
of_GetMethod
 n_cst_lvsrv_datasource 1079
 n_cst_tvsrv_levelsource 1314
of_GetMicrohelp 570
of_GetMinimum 449
of_GetMinMaxPoints 1181
of_GetMinObjectSize 473
of_GetMRUItem 1130
of.GetName 540
of.GetNext 1146
of.GetNextLevel 337
of.GetNotifyConnection 944
of.GetNotifyPreTitle 945
of.GetNotifySeverity 945
of.GetNotifyWho 946
of.GetObjectInformation 881
of.GetObjects
 n_cst_dwsrv 680
 n_cst_tvsrv_levelsource 1315
 u_lvs 222
 u_tvs 337
of.GetOtherSaveObjects 749
of.GetPageInputField 279
of.SetParentPosition 642
of.SetParentWindow
 n_cst_dropdown 643
 n_ds 501, 505
 u_base 370
 u_cb 113
 u_cbx 116
 u_ddlb 122
 u_ddplb 129
 u_dw 159
 u_em 183
 u_gr 189
 u_hpb 192
 u_hsb 195
 u_htb 198
 u_lb 201
 u_lvs 223
 u_mle 241
 u_oc 248
 u_p 252
 u_pb 255
 u_phl 258
 u_plb 261
 u_rb 265
 u_rte 279
 u_sle 295
 u_st 298

u_sth 301
 u_tab 308
 u_tvs 338
 u_vpb 354
 u_vsb 357
 u_vtb 360
 of_GetPctComplete 450
 of_GetPhysicalMemory
 n_cst_platform 1160
 n_cst_platformwin32 1168
 of_GetPictureColumn
 n_cst_lvsrv_datasource 1080
 n_cst_tvsvr_levelsouce 1317
 of_GetPictureSize 847
 ofGetPosition 450
 of_GetPowerObjectParm 521
 of_GetPredefinedSource 946
 of_GetPredefinedSourceType 947
 of_GetPrev 1146
 ofGetPropertyInfo
 n_cst_dwsrv 681
 n_cst_dwsrv_dropdownsearch 694
 n_cst_dwsrv_filter 705, 720
 n_cst_dwsrv_linkage 750
 n_cst_dwsrv_multitable 793
 n_cst_dwsrv_printpreview 805
 n_cst_dwsrv_querymode 819
 n_cst_dwsrv_report 848
 n_cst_dwsrv_reqcolumn 871
 n_cst_dwsrv_resize 882
 n_cst_dwsrv_rowmanager 900
 n_cst_dwsrv_rowselection 912
 n_cst_dwsrv_sort 926
 u_calculator 386
 u_calendar 416
 of_GetQueryCols 819
 of_GetRefreshRate 1403
 of_GetRegisterable
 n_cst_dwsrv_dropdownsearch 695
 n_cst_dwsrv_filter 706
 n_cst_dwsrv_multitable 794
 n_cst_dwsrv_reqcolumn 872
 n_cst_dwsrv_resize 883
 n_cst_dwsrv_sort 927
 u_calculator 387
 u_calendar 416
 of_GetRegisterableColumn 795
 of_GetRegisterableTable 796
 of_GetRegistered
 n_cst_dwcache 654
 n_cst_dwsrv_dropdownsearch 696
 n_cst_dwsrv_linkage 751
 n_cst_dwsrv_multitable 796
 n_cst_dwsrv_reqcolumn 872
 n_cst_dwsrv_resize 883
 n_cst_mru 1130
 n_cst_tmgsingle 1247
 n_cst_trregistration 1276
 u_calculator 388
 u_calendar 417
 of_GetRegisteredl
 n_cst_tmgtmultiple 1238
 of_GetRegisteredStyle
 u_calculator 388
 u_calendar 418
 of_GetResetCriteria 820
 of_GetRestoreRow 901
 of_GetRetrieveArgs 1317
 of_GetRetrieveOnDisabled 820
 of_GetRoot 1262
 of_GetRow 223
 of_GetRuler 806
 of_GetSaturdayColor 418
 of_GetSaveSound 751
 of_GetSaveStatus 40
 of_GetSections 1015
 of_GetSelected
 u_lb 202
 u_plb 262
 of_GetSelectedPictureColumn 1318
 of_GetSeparator 982
 of_GetSheetCount 1375
 of_GetSheets 1375
 of_GetSheetsByClass 1376
 of_GetSheetsByTitle 1377
 of_GetSort 927
 of_GetSqlState 541
 of_GetStartPageNumber 280
 of_GetStatePictureColumn
 n_cst_lvsrv_datasource 1080
 n_cst_tvsvr_levelsouce 1318
 of_GetStep 451

Index

of_GetStringParm 521
of_GetStyle
 n_cst_dwsrv_filter 707
 n_cst_dwsrv_linkage 752
 n_cst_dwsrv_rowselection 913
 n_cst_dwsrv_sort 928
 n_cst_error 947
of_GetSundayColor 419
of_GetSystemDirectory
 n_cst_platform 1161
 n_cst_platformwin32 1169
of_GetSystemSetting 643
of_GetTextColor 451
of_GetTextSize
 n_cst_platform 1161
 n_cst_platformwin32 1169
of_GetTextSizePos 849
of_GetTextStyle 280
of_GetTimeOut 948
of_GetTimer 1403
of_GetTimerInterval 1404
of_GetToken 1217
of_GetToolbarItemOrder 1357
of_GetToolbarItemSpace 1358
of_GetToolbarItemVisible 1359
of_GetToolbars 1359
of_GetToolbarTitles 1360
of_GetTrace 541
of_GetTransObject
 n_cst_lvsrv_datasource 1081
 n_cst_tvsrv_levelsource 1319
of.GetType 1043
of.GetTypeToProcess 1044
of.GetUnattended 949
of.GetUndoLevels 850
of.GetUpdateable 160
of.GetUpdateBottomUp 753
of.GetUpdateObjects
 u_base 371
 u_tab 309
 w_master 41
of.GetUpdateRequestor
 n_cst_luw 1044
 u_lvs 224
 u_tvs 339
of.GetUpdatesPending 753
of.GetUpdateStyle
 n_cst_dwsrv_linkage 754
 n_cst_tvsrv_levelsource 1319
of.GetUseColLinks 754
of.GetUseDisplay 928
of.GetUser
 n_cst_error 949
 n_cst_winsrv_statusbar 1404
of.GetUserID
 n_cst_appmanager 570
 n_cst_platform 1162
 n_cst_platformwin32 1170
of.GetUserIniFile 571, 1131
of.GetUserKey 571, 1131
of.GetUserThreshold 1405
of.GetValue 755
of.GetVersion 572
of.GetVerticalPointer 474
of.GetVisibleOnly
 n_cst_dwsrv_filter 707
 n_cst_dwsrv_sort 929
of.GetWidth 682
of.GetWindow 1360
of.GetWindowsDirectory
 n_cst_platform 1162
 n_cst_platformwin32 1171
of.GetWindowText
 n_cst_platform 1163
 n_cst_platformwin32 1171
of.GetXPosColumn 1081
of.GetYPosColumn 1082
of.GetZoom 806
of.GlobalReplace 1218
of.Gregorian 619
of.Hours 620
of.Icon 608
of.IncludeComputes
 n_cst_dwsrv_find 721
of.IncludeFile
 n_cst_filesrv 983
of.Increment 452
of.Init
 n_cn 485
 n_tr 542

of_Initialize 756
 n_cst_dwsrv_find 722
 n_cst_rtefind 1190
 of_InitSecurity 1194
 of_InsertDocument 281
 of_InsertItem
 n_cst_lvsvr_datasource 1082
 n_cst_tvsvr_levelsouce 1320
 u_lvs 224, 225
 u_tvs 339, 340
 of_InsertPicture 282
 of_InsertRow 901
 of_Integer 609
 of_InvertSelection 913
 of_IsAllowFindDlg 722
 of_IsAllowReplaceDlg 723
 of_IsAllowSqlcaDestroy 1277
 of_IsAlpha 1219
 of_IsAlphaNum 1220
 of_IsAlwaysRedraw 419
 of_IsAlwaysValidate
 n_cst_luw 1045
 u_base 371
 u_lvs 226
 u_tab 309
 u_tvs 341
 w_master 41
 of_IsAncestorClass 1117
 of_IsArithmeticOperator 1220
 of_IsAutoDisable
 n_cst_tmgsingle 1248
 of_IsAutoDisable1
 n_cst_tmgmultiple 1239
 of_IsAutoReset 452
 of_IsCloseOnClick
 u_calculator 389
 u_calendar 420
 of_IsCloseOnDClick 420
 of_IsColumnHeader 1105
 of_IsCompareObjectAssigned
 n_cst_linkedListbase 1021
 of_IsComparisonOperator 1221
 of_IsConfirmOnDelete
 n_cst_dwsrv_linkage 755
 n_cst_lvsvr_datasource 1083
 n_cst_tvsvr_levelsouce 1321
 of_IsConfirmOnRowChange 756
 of_IsConnected 543
 of_IsDateType 421
 of_IsDestructorDestroy 1277
 of_IsDisabled
 n_cst_tmgsingle 1248
 of_IsDisabled1
 n_cst_tmgmultiple 1239
 of_IsDropDown
 u_calculator 390
 of_IsDuplicatesAllowed 1033
 of_IsEmpty 1221
 n_cst_linkedListbase 1022
 of_IsEventDefined 1118
 of_IsEventImplemented 1119
 of_IsExclude 1106
 of_IsFormat 1222
 of_IsFunctionDefined 1120
 of_IsHolidayBold 422
 of_IsInArray 1111
 of_IsInitialValue
 u_calculator 390
 u_calendar 422
 of_IsInList 1034
 of_IsKey 756
 of_IsLeapYear 620
 of_IsLinked 757
 of_IsLogOpen 637
 of_IsLower 1222
 of_IsMarkedDayBold 423
 of_IsNumericType 391
 of_IsOngoingFind
 n_cst_dwsrv_find 723
 n_cst_rtefind 1191
 of_IsOperator 392
 of_IsPredefined 1406
 of_IsPrintable 1223
 of_IsPropertyOpen 812
 of_IsPunctuation 1224
 of_IsRecursiveLevel 1322
 of_IsRegistered
 n_cst_dwcache 656
 n_cst_dwsrv_dropdownsearch 696
 n_cst_dwsrv_multitable 797
 n_cst_dwsrv_reqcolumn 873
 n_cst_dwsrv_resize 884

Index

n_cst_mru 1132
n_cst_tmgsingle 1240, 1249
n_cst_trregistration 1278
u_calculator 392
u_calendar 423
of_IsRegistryAvailable 572
of_IsRestoreApp 588
of_IsRestoreUser 588
of_IsRMBMenu
 u_lvs 227
 u_tvs 342
of_IsRoot
 n_cst_dwsrv_linkage 757
 u_dw 160
of_IsSaturdayBold 424
of_IsSelfUpdatingObject 1045
of_IsSharedProperty 160
of_IsSorted 1034
of_IsSpace 1224
of_IsSundayBold 424
of_IsSyncOnKeyChange 758
of_IsUpdateable
 n_ds 501
 u_base 372
 u_dw 161
 u_lvs 227
 u_tab 310
 u_tvs 342
 w_master 42
of_IsUpdateOnRowChange 758
of_IsUpper 1225
of_IsValid 621
of_IsWeekDay 622
of_IsWeekEnd 622
of_IsWhiteSpace 1225
of_ItemFocusChanged 759
of_Julian 623
of_JulianDayNumber 624
of_KeybRowSelect 914
of_KeybRowSelectExt 915
of_KeyChanged 759
of_KeySync 760
of_LastDayOfMonth 624
of_LastPage 806
of_LastPos 1226
of_LButtonUp
 u_st_splitbar 474
of_LeftTrim 1227
of_LinkDetail 761
of_LinkTo 761
of_Load 821
of_LoadMRU 1132
of_LoadPredefinedMsg 950
of_LogonDlg 573
of_Message
 n_cst_debug 638
 n_cst_error 950
of_MessageBox
 n_base 552
 n_cn 486
 n_ds 502
 n_err 514
 n_ms 517
 n_msg 522
 n_pl 527
 n_tr 543
 n_trp 548
 u_base 372
 u_cb 113
 u_cbx 116
 u_ddlb 123
 u_ddplb 130
 u_dw 161
 u_em 184
 u_gr 189
 u_hpb 192
 u_hsb 195
 u_htb 198
 u_lb 202
 u_lvs 227
 u_mle 241
 u_oc 249
 u_p 252
 u_pb 255
 u_phl 258
 u_plb 262
 u_rb 265
 u_rte 282
 u_sle 295
 u_st 298

u_sth 301
 u_tab 310
 u_tvs 343
 u_vpb 354
 u_vsb 357
 u_vtb 360
 w_master 42
 of_MilliSecsAfter 625
 of_Modify
 n_cst_dwsrv 682
 n_cst_winsrv_statusbar 1407
 of_MonthsAfter 626
 of_MouseMove 475
 of_Open
 n_cst_selection 1198
 n_cst_winsrv_statusbar 1407
 of_OpenDocument 284
 of_OpenLog 638
 of_OpenProperty 813
 of_OpenPropertyService 813
 of_OpenSQLSpy 1207
 of_OSType 609
 of_PadLeft 1227
 of_PadRight 1228
 of_Page 807
 of_PageAcross 807
 of_PageCount 808
 of_PageCountAcross 808
 of_PageSetupDlg 1163
 of_Parse 1203
 of_ParseArgs 1322
 of_ParseCols 1323
 of_ParsePath 984
 of_ParseSortAttrib 930
 of_ParseToArray 1228
 of_PerformMath 393
 of_PlaySound
 n_cst_platform 1163
 n_cst_platformwin32 1172
 of_Pop 1212
 of_Populate
 u_lvs 228
 u_tvs 344
 of_PopulateDDDw 684
 of_Position 645
 of_PostUpdate

n_cst_dwsrv_linkage 762
 n_cst_luw 1046
 n_ds 503
 u_base 373
 u_dw 162
 u_lvs 229
 u_tab 311
 u_tvs 344
 w_master 43
 of_PrepPrint 850
 of_PreUpdate 1046
 of_PrintDlg
 n_cst_platform 1164
 n_cst_platformwin32 1172
 of_PrintLog 638
 of_PrintReport 851
 of_PrintTree 1348
 of_ProcessLog 953
 of_ProcessMessage 953
 of_ProcessMessageSubstitution 954
 of_ProcessNotify 954
 of_Push 1213
 of_Put 1176
 of_Quote 1229
 of_ReadKeys 1134
 of_ReadMRUKeys 1133
 of_RecalcInterval 1240
 of_RedirectFocus
 u_calculator 393
 u_calendar 425
 of_Redraw 475
 of_Refresh
 n_cst_dwcache 657
 n_cst_dwsrv_linkage 762
 n_cst_lvsrv_datasource 1084
 of_RefreshDDDWs 685
 of_Register
 n_cst_dwcache 657, 659
 n_cst_dwsrv_dropdownsearch 697
 n_cst_dwsrv_linkage 763
 n_cst_dwsrv_multitable 797
 n_cst_dwsrv_resize 885
 n_cst_lvsrv_datasource 1084
 n_cst_mru 1134
 n_cst_resize 1182
 n_cst_tmgsingle 1241, 1249

n_cst_trregistration 1278
n_cst_tvsrv_levelsource 1324
n_cst_winsrv_statusbar 1408
u_calculator 394
u_calendar 425
u_st_splitbar 476
of_RegisterArgs 663
of_RegisterDataSource
 n_cst_lvsrv_datasource 1091
 n_cst_tvsrv_levelsource 1332
of_RegisterPredefined 1409
of_RegisterReportColumn 1092
of_RegisterSkipColumn 873
of_RelativeDateTime 627
of_RelativeMonth 628
of_RelativeYear 628
of_Remove
 n_cst_linkedListbase 1022
 n_cst_list 1035
 n_cst_tree 1263
of_RemoveChildren 1333
of_RemoveColumn 697
of_RemoveDirectory
 n_cst_filesvr 984
 n_cst_filesvrwin32 1007
of_RemoveHead
 n_cst_linkedListbase 1023
 n_cst_list 1036
of_RemoveNonPrint 1230
of_RemoveTail
 n_cst_linkedListbase 1023
of_RemoveWhiteSpace 1230
of_Replace 724
of_Reset
 n_cst_dwsrv_linkage 764
 n_cst_lvsrv_datasource 1093
 n_cst_mru 1135
 n_cst_tvsrv_levelsource 1334
 u_calculator 396
 u_calendar 428
 u_dw 163
 u_lvs 229
 u_progressbar 453
 u_tvs 345
of_ResetArguments 764
of_ResetList 1093
 of_ResetMaster 764
 of_ResetTree 1334
 of_ResetUndo 852
 of_ResetUpdate
 n_cst_dwsrv_linkage 765
 n_cst_lvsrv_datasource 1093
 n_cst_tvsrv_levelsource 1335
of_Resize
 n_cst_dwsrv_resize 889
 n_cst_resize 1184
of_Restore
 n_cst_apppreference 589
 n_cst_mru 1136
 n_cst_winsrv_preference 1361
of_RestoreApp 591
of_RestoreFocusPoint 1410
of_RestoreMenu 1362
of_RestorePositiveNumber 1363
of_RestoreUpdateSettings 798
of_RestoreUser 592
of_Retrieve
 n_cst_dwsrv_linkage 766
 n_cst_lvsrv_datasource 1094
 n_cst_tvsrv_levelsource 1335
 n_ds 503
 u_dw 163
 u_lvs 230
 u_tvs 345
 of_RetrieveDetails 766
of_RightTrim 1231
of_Rollback 544
of_RowSelect 915
of_RowSelectExt 916
of_RowSelectMulti 917
of_RowSelectSingle 917
of_Save
 n_cst_apppreference 592
 n_cst_dwsrv_linkage 767
 n_cst_dwsrv_querymode 822
 n_cst_luw 1047
 n_cst_mru 1136
 n_cst_winsrv_preference 1364
of_SaveApp 594
of_SaveMenu 1366
of_SaveUser 595
of_ScrollDetails 768

of_Search 1264
 of_SearchChild 1287
 of_SearchItem 698
 of_SecondsAfter 629
 of_SelectedCount 918
 of_SelectText 725
 of_SendMessage
 m_master 72
 n_cst_menu 1111
 of_SetAllowFindDlg 726
 of_SetAllowReplaceDlg 726
 of_SetAllowSqlcaDestroy 1279
 of_SetAlwaysOnTop
 n_cst_debug 639
 n_cst_sqlspy 1207
 of_SetAlwaysRedraw 428
 of_SetAlwaysValidate
 n_cst_luw 1048
 u_base 374
 u_lvs 230
 u_tab 312
 u_tvs 346
 w_master 44
 of_SetAppend 504
 of_SetAppIniFile
 n_cst_appmanager 574
 of_SetAppKey 574, 597
 of_SetAppPreference
 n_cst_appmanager 575
 of_SetArguments 769
 of_SetAutoDisable
 n_cst_tmgsingle 1250
 of_SetAutoDisableI
 n_cst_tmgmultiple 1242
 of_SetAutoReset 453
 of_SetAutoRollback 545
 of_SetBackColor 454
 of_SetBackground 853
 of_SetBalance 1269
 of_SetBar 1410
 of_SetBarAutoClear 1411
 of_SetBarColor 477
 of_SetBarFillColor 1412
 of_SetBarFillStyle 1413
 of_SetBarMax 1413, 1414
 of_SetBarMoveColor 477
 of_SetBarOffsetX 1414
 of_SetBarOffsetY 1415
 of_SetBarPosition 1415
 of_SetBarStep 1416
 of_SetBarStyle 1411
 of_SetBarTextColor 1417
 of_SetBarWidth 1417
 of_SetBase
 n_ds 504
 u_dw 164
 u_lvs 231
 u_tvs 346
 w_master 44
 of_SetBatchMode 1208
 of_SetBeep 955
 of_SetBorder 855
 of_SetBorderType 1418
 of_SetCache
 n_cst_lvsrv_datasource 1094
 n_cst_tvsrv_levelsouce 1336
 of_SetCloseOnClick
 u_calculator 397
 u_calendar 429
 of_SetCloseOnDClick 429
 of_SetColor 857
 of_SetColumnDisplayStyle 685
 of_SetColumnHeader
 n_cst_dwsrv_sort 930
 n_cst_lvsrv_sort 1106
 of_SetColumnNameSource 685
 of_SetCompare
 n_cst_linkeditlistbase 1023
 n_cst_tree 1264
 of_SetConfirmonDelete
 n_cst_dwsrv_linkage 769
 n_cst_dwsrv_rowmanager 902
 n_cst_lvsrv_datasource 1095
 n_cst_tvsrv_levelsouce 1337
 of_SetConfirmOnRowChange 770
 of_SetContinuousPages 284
 of_SetCopyright 575
 of_SetCreationDateTime
 n_cst_filesvr 985
 n_cst_filesvrwin32 1008
 of_SetCurrentState 1378

of_SetCustomUpdate
 n_cst_dwsrv_linkage 771
 n_cst_tvsrv_levelsource 1337
of_SetData 1146
of_SetDataSource
 u_lvs 231
 u_tvs 347
of_SetDate
 u_calendar 430
of_SetDateFormat
 u_calendar 431
of_SetDBErrorMsg
 n_cst_luw 1049
 w_master 45
of_SetDebug 576
of_SetDefaultBackColor 858
of_SetDefaultBorder 859
of_SetDefaultCharset 860
of_SetDefaultColor 860
of_SetDefaultFontFace 861
of_SetDefaultFontSize 861
of_SetDefaultHeaderSuffix 686
of_SetDeleteStyle
 n_cst_dwsrv_linkage 771
 n_cst_tvsrv_levelsource 1338
of_SetDelimiter 1137
of_SetDestructorDestroy 1279
of_SetDisabled
 n_cst_tmgsingle 1251
of_SetDisabledl
 n_cst_tmgtmultiple 1242
of_SetDisplayItem
 n_cst_dwsrv 686
 n_cst_lvsrv 1058
 n_cst_tvsrv 1288
of_SetDisplayStyle 454
of_SetDisplayUnits
 n_cst_dwsrv 687
 n_cst_lvsrv 1058
 n_cst_tvsrv 1288
of_SetDoubleParm 523
of_SetDropDown
 u_calculator 398
 u_calendar 431
of_SetDropDownCalculator
 u_dw 164
 u_em 185
of_SetDropDownCalendar
 u_dw 165
 u_em 186
of_SetDropDownSearch 165
of_SetDuplicatesAllowed 1037
of_SetDwCache 576
of_SetDWProperty 640
of_SetEnabled
 n_cst_dwsrv_printpreview 809
 n_cst_dwsrv_querymode 823
of_SetError 577
of_SetExclude
 n_cst_dwsrv_filter 708
 n_cst_dwsrv_sort 931
 n_cst_lvsrv_sort 1107
of_SetFileArchive 985
of_SetFileAttributes
 n_cst_filesrv 986
 n_cst_filesvrwin32 1009
of_SetFileHidden 986
of_SetFileName 285
of_SetFile_READONLY 987
of_SetFileSystem 987
of_SetFillColor 455
of_SetFillStyle 456
of_SetFilter
 n_cst_dwsrv_filter 708
 u_dw 166
of_SetFind
 u_dw 167
 u_rte 285
of_SetFindWindowClasses 1164
of_SetFocusOnRequestor
 u_calculator 398
 u_calendar 432
of_SetFont
 n_cst_dwsrv_report 862
 u_progressbar 456
of_SetFontBold 458
of_SetFontCharSet 459
of_SetFontFace 459
of_SetFontFamily 460

of_SetFontItalic 460
 of_SetFontPitch 461
 of_SetFontSize 461
 of_SetFontUnderline 462
 of_SetFrame 577
 of_SetGapWidth 1418
 of_SetGDI 1419
 of_SetGDIThreshold 1419
 of_SetGDIWidth 1420
 of_SetHelpFile 578
 of_SetHoliday 433
 of_SetHolidayBold 433
 of_SetHolidayColor 434
 of_SetHorizontalPointer 477
 of_SetIgnoreFileExists 286
 of_SetInitialValue
 u_calculator 399
 u_calendar 434
 of_SetItem 687
 of_SetItemAttributes
 n_cst_lvsrv_datasource 1096
 n_cst_tvsrv_levelsouce 1338
 of_SetItemCount 1137
 of_SetKey 1147
 of_SetKeyboard 918
 of_SetKeyCols 772
 of_SetKeyValue 1232
 of_SetKeyValues 772
 of_SetLastAccessDate
 n_cst_filesv 988
 n_cst_filesvwin32 1010
 of_SetLastWriteDateTime
 n_cst_filesv 988
 n_cst_filesvwin32 1011
 of_SetLineEnding 1016
 of_SetLinkage 167
 of_SetLogFile
 n_cst_error 955
 n_cst_sqllspy 1208
 of_SetLogFileStyle 956
 of_SetLogicalUnitOfWork
 u_base 374
 u_lvs 232
 u_tab 312
 u_tvs 347
 w_master 46
 of_SetLogidAndPassword 545
 of_SetLogo 578
 of_SetLogSeverity 957
 of_SetMarkedDay 435
 of_SetMarkedDayBold 436
 of_SetMarkedDayColor 436
 of_SetMaster 773
 of_SetMaximum 462
 of_SetMem 1421
 of_SetMemThreshold 1421
 of_SetMemWidth 1422
 of_SetMenuItems 1366
 of_SetMessageText 463
 of_SetMicrohelp 579
 of_SetMinimum 464
 of_SetMinObjectSize 478
 of_SetMinSize
 n_cst_dwsrv_resize 890
 n_cst_resize 1184
 of_SetMRU 579
 of_SetMultiple 530
 of_SetMultiTable
 n_ds 505
 u_dw 168
 of_SetName 546
 of_SetNext 1148
 of_SetNotifyConnection 957
 of_SetNotifyPreTitle 958
 of_SetNotifySeverity 958
 of_SetNotifyWho 959
 of_SetOrigSize
 n_cst_dwsrv_resize 890
 n_cst_resize 1185
 of_SetOtherSaveObjects 774
 of_SetOverlayPicture
 n_cst_lvsrv_datasource 1079
 n_cst_tvsrv_levelsouce 1316
 of_SetOverLayPictureColumn
 n_cst_lvsrv_datasource 1096
 n_cst_tvsrv_levelsouce 1339
 of_SetPageInputField 286
 of_SetParent
 m_dw 61
 m_edit 63
 m_oc 74

m_tvs 76
 m_view 66
of_SetPictureColumn
 n_cst_lvsrv_datasource 1097
 n_cst_tvsrv_levelsouce 1339
of_SetPosition 464
of_SetPosSize 1367
of_SetPowerObjectParm 523
of_SetPredefinedSource 960
of_SetPreference
 w_master 46
of_SetPrev 1148
of_SetPrint 348
of_SetPrintPreview
 n_ds 506
 u_dw 168
of_SetProperty 169
of_SetQueryCols 823
of_SetQuerymode 169
of_SetRecursive 1340
of_SetRedraw 774
of_SetRefreshRate 1422
of_SetRelativeZoom 866
of_SetReport
 n_ds 506
 u_dw 170
of_SetReqColumn 170
of_SetRequestor
 n_cst_apppreference 598
 n_cst_dropdown 645
 n_cst_dwsrv 688
 n_cst_dwsrv_resize 891
 n_cst_luw 1049
 n_cst_lvsrv 1059
 n_cst_rtefind 1192
 n_cst_tmgsingle 1251
 n_cst_tvsrv 1289
 n_cst_winsrv 1352
 u_calculator 399
 u_calendar 437
of_SetRequestorl
 n_cst_tmglemultiple 1243
of_SetResetCriteria 824
of_SetResize
 u_base 375
 u_dw 171
 u_tab 313
 w_master 47
of_SetRestoreApp 598
of_SetRestoreRow 902
of_SetRestoreUser 599
of_SetRetrieveOnDisabled 825
of_SetRMBMenu
 u_lvs 232
 u_tvs 348
of_SetRowManager 171
of_SetRowSelect 172
of_SetRuler 810
of_SetSaturdayBold 438
of_SetSaturdayColor 438
of_SetSaveSound 775
of_SetSecurity
 n_cst_appmanager 579
 n_cst_security 1196
of_SetSelectedPictureColumn 1341
of_SetSharedProperty 172
of_SetSheetManager 16
of_SetSingle 531
of_SetSort
 n_cst_dwsrv_sort 931
 u_dw 173
 u_lvs 232
of_SetSorted 1037
of_SetSQLSpy 640
of_SetStartPageNumber 287
of_SetStatePictureColumn
 n_cst_lvsrv_datasource 1097
 n_cst_tvsrv_levelsouce 1342
of_SetStatusBar 16
of_SetStep 465
of_SetStringParm 524
of_SetStyle
 n_cst_dwsrv_filter 709
 n_cst_dwsrv_linkage 775
 n_cst_dwsrv_rowselection 919
 n_cst_dwsrv_sort 932
 n_cst_error 961
of_SetSundayBold 439
of_SetSundayColor 439
of_SetSyncOnKeyChange 776
of_SetTextColor 465
of_SetTextStyleBold 287

of_SetTextStyleItalic 288
 of_SetTextStyleStrikeout 288
 of_SetTextStyleSubscript 289
 of_SetTextStyleSuperscript 289
 of_SetTextStyleUnderline 290
 of_SetTimeOut 961
 of_SetTimer 1423
 of_SetTimerFormat 1423
 of_SetTimerInterval 1424
 of_SetTimerWidth 1424
 of_SetToNullIfEmpty 874
 of_SetToolbarItemOrder 1367
 of_SetToolbarItemSpace 1368
 of_SetToolbarItemVisible 1369
 of_SetToolbars 1369
 of_SetToolbarTitles 1370
 of_SetTrace 546
 of_SetTransObject
 n_cst_dwsrv_linkage 776
 n_ds 506
 u_dw 173
 of_SetTrRegistration 580
 of_SetTypeToProcess 1050
 of_SetUnattended 962
 of_SetUndo
 n_cst_lvsvr_datasource 1098
 n_cst_tvsrv_levelsouce 1342
 of_SetUpdateable
 n_ds 507
 u_base 375
 u_dw 174
 u_lvs 233
 u_tab 313
 u_tvs 348
 w_master 47
 of_SetUpdateBottomUp 777
 of_SetUpdateObjects
 u_base 376
 u_tab 314
 w_master 47
 of_SetUpdateOnRowChange 778
 of_SetUpdateRequestor
 n_cst_dwsrv_linkage 778
 n_cst_luw 1050
 n_ds 507
 u_base 376
 u_dw 174
 u_lvs 233
 u_tab 313
 u_tvs 348
 w_master 47
 u_dw 174
 u_lvs 233, 314
 u_tvs 349
 w_master 48
 of_SetUpdateStyle
 n_cst_tvsrv_levelsouce 779, 1343
 of_SetUseColLinks 780
 of_SetUseDisplay 933
 of_SetUser
 n_cst_error 962
 n_cst_winsrv_statusbar 1425
 n_tr 546
 of_SetUseRegistry 1138
 of_SetUserID 580
 of_SetUserIdAndPassword 547
 of_SetUserIniFile 1138
 n_cst_appmanager 581
 n_cst_apppreference 600
 of_SetUserKey 1139
 n_cst_appmanager 581
 n_cst_apppreference 601
 of_SetUserThreshold 1425
 of_SetUserWidth 1426
 of_SetValue 400
 of_SetValueOnRequestor 401
 of_SetVersion 582
 of_SetVerticalPointer 479
 of_SetVisibleOnly
 n_cst_dwsrv_filter 710
 n_cst_dwsrv_sort 933
 of_SetWindow 1370
 of_SetXPosColumn 1099
 of_SetYPosColumn 1099
 of_SetZoom 810
 of_ShiftBand 866
 of_Sort 934
 of_SortDirList 989
 of_SortMRU 1140
 of_Splash 582
 of_SQLPreviewType 610
 of_SQLSyntax 1209
 of_StoreUpdateSettings 799
 of_String 611
 of_Time 614
 of_ToolbarAlignment 614
 of_ToolbarExists 1112

Index

of_Traverse 1265
of_TriggerEvent
 n_cst_dwsrv_linkage 780
 n_cst_menu 1112
of_Trim 1232
of_TypeOf 1185
of_UnDelete 903
of_Undo 867
of_UndoModified 781
of_Unlink 781
of_UnlinkDetail 781
of_UnRegister
 n_cst_dwcache 663
 n_cst_dwsrv_dropdownsearch 698
 n_cst_dwsrv_linkage 782
 n_cst_dwsrv_multitable 800
 n_cst_dwsrv_reqcolumn 875
 n_cst_dwsrv_resize 892
 n_cst_mru 1140
 n_cst_resize 1186
 n_cst_tmgsingle 1252
 n_cst_trregistration 1280
 n_cst_winsrv_statusbar 1426
 u_calculator 402
 u_calendar 440
of_UnRegister1
 n_cst_tmgsmultiple 1243
of_UnRegisterPredefined 1427
of_UnRegisterReportColumn 1100
of_UnRegisterst
of_Update
 n_cst_dwsrv_linkage 783
 n_cst_dwsrv_multitable 800
 n_cst_luw 1051
 n_cst_lvsrv_datasource 1101
 n_cst_tvsrv_levelsource 1344
 n_ds 508
 u_base 377
 u_dw 175
 u_lvs 234
 u_tab 315
 u_tvs 349
 w_master 49
of_UpdateBottomUp 784
of_UpdateBottomUpAndTopDown 784
of_UpdateChecks 49
of_UpdateCustom
 n_cst_dwsrv_linkage 785
 n_cst_tvsrv_levelsource 1345
of_UpdateObjectData 480
of_UpdateOnRowChange 786
of_UpdatePredefined 1428
of_UpdatePrep 786
 n_cst_luw 1052
 n_ds 510
 u_base 377
 u_dw 177
 u_lvs 235
 u_tab 315
 u_tvs 350
 w_master 50
of_UpdateSkipString 875
of_UpdatesPending
 n_cst_luw 1052
 n_ds 511
 u_base 378
 u_dw 177
 u_lvs 235
 u_tab 316
 u_tvs 351
 w_master 51
of_UpdateTopDown 787
of_UpdateTopDownAndBottomUp 788
of_UpdateVisuals 466
 n_cst_mru 1141
of_UsesDisplayValue 934
of_Validation
 n_cst_dwsrv_linkage 789
 n_cst_luw 1053
 n_ds 511
 u_base 379
 u_dw 178
 u_lvs 236
 u_tab 317
 u_tvs 351
 w_master 52
of_Wait 630
of_WeekNumber 632
of_WeeksAfter 632
of_WindowState 615

of_WordCap 1233
 of_WriteKeys 1142
 of_WriteMRUKeys 1141
 of_YearsAfter 633
 OLEControl 243
 OLEObject object 525
 OLEStorage object 525
 OLEStream object 526
 online Help
 accessing 6
 of_GetHelpFile 569
 of_SetHelpFile 578
 Open event 21

P

PEAT sample application 7
 pfc_AcceptText
 u_base 366
 u_dw 139
 u_lvs 209
 u_tab 304
 u_tvs 323
 w_master 22
 pfc_AddAll
 u_lvs 209
 u_tvs 323
 pfc_AddItem
 u_lvs 209
 u_tvs 324
 pfc_AddRow
 n_cst_dwsrv_rowmanager 895
 u_dw 140
 pfc_Apply 55
 pfc_BeginTran 22
 pfc_Cancel 55
 pfc_Cascade
 n_cst_winsrv_sheetmanager 1373
 w_frame 13
 pfc_Clear
 u_ddlb 120
 u_ddplb 127
 u_dw 140
 u_em 181
 u_mle 239
 pfc_Clicked
 n_cst_dwsrv_linkage 732, 733
 n_cst_dwsrv_rowselection 907
 n_cst_dwsrv_sort 922
 pfc_Close
 n_cst_appmanager 560
 n_cst_apppreference 586
 n_cst_mru 1125
 w_master 22
 pfc_ColumnClick 1103
 pfc_ConnectionBegin 560
 pfc_ConnectionEnd 561
 pfc_ControlGotFocus
 w_master 22
 w_sheet 58
 pfc_Copy
 u_ddlb 120
 u_ddplb 127
 u_dw 140
 u_em 181
 u_mle 239
 u_oc 245
 u_rte 270
 u_sle 293
 pfc_Cut
 u_ddlb 121
 u_ddplb 128
 u_dw 140
 u_em 181
 u_mle 239
 u_oc 245
 u_rte 271
 u_sle 293
 pfc_DBError 23
 pfc_DDCalculator
 u_dw 141
 u_em 181
 pfc_DDCalendar
 u_dw 141
 u_em 182
 pfc_Debug 142
 pfc_Decode
 n_cst_mru 1125

pfc_Default 56
pfc_Delete
 u_lvs 210
 u_tvs 324
pfc_DeleteItem
 n_cst_lvsrv_datasource 1063
 n_cst_tvsrv_levelsource 1293
pfc_DeleteRow
 n_cst_dwsrv_linkage 733
 n_cst_dwsrv_rowmanager 895
 u_dw 142
pfc_Descendant
 u_dw 142
 w_master 23
pfc_DropDown
 u_calculator 384
 u_calendar 409
pfc_EditChanged 691
pfc>EditObject 245
pfc_Encode
 n_cst_mru 1126
pfc_EndLabelEdit
 n_cst_lvsrv_datasource 1063
 n_cst_tvsrv_levelsource 1294
pfc_EndTran
 w_master 23
pfc_Exit 562
pfc_FilterDlg
 n_cst_dwsrv_filter 702
 u_dw 142
pfc_FindDlg
 n_cst_dwsrv_find 713
 n_cst_rtefind 1188
 u_dw 142
 u_rte 271
pfc_FindNext
 n_cst_dwsrv_find 713
 n_cst_rtefind 1188
pfc_FirstPage
 u_dw 143
 u_rte 271
pfc_Help 24
pfc_Hide event 1384
pfc_Idle 562
pfc_InsertFile 271

pfc_InsertItem
 n_cst_lvsrv_datasource 1064
 n_cst_tvsrv_levelsource 1295
 u_lvs 210
 u_tvs 324
pfc_InsertObject 246
pfc_InsertPicture 272
pfc_InsertRow
 n_cst_dwsrv_linkage 734
 n_cst_dwsrv_rowmanager 896
 u_dw 143
pfc_InvertSelection
 u_lb 201
 u_lvs 211
 u_plb 260
pfc_ItemChanged 734
pfc_ItemExpanding 1296
pfc_ItemFocusChanged
 n_cst_dwsrv_dropdownsearch 691
 n_cst_dwsrv_linkage 734
pfc_LastPage
 u_dw 143
 u_rte 272
pfc_Layer
 n_cst_winsrv_sheetmanager 1374
 w_frame 13
pfc_LButtonDown 907
pfc_LButtonUp 908
pfc_Logon 562
pfc_MessageRouter 24
pfc_MicroHelp
 w_frame 13
 w_master 25
 w_sheet 58
pfc_MinimizeAll
 n_cst_winsrv_sheetmanager 1374
 w_frame 14
pfc_Move event 1384
pfc_MRUClicked 25
pfc_MRUProcess 25
pfc_MRURestore 26
pfc_MRUSave 27
pfc_New
 u_lvs 212
 u_tvs 326
 w_master 27

pfc_NextDay 409
 pfc_NextMonth 410
 pfc_NextPage
 u_dw 143
 u_rte 272
 pfc_NextWeek 410
 pfc_Open
 n_cst_appmanager 563
 n_cst_apppreference 587
 n_cst_mru 1126
 u_rte 272
 w_master 27
 pfc_OpenObject 246
 pfc_Operators
 n_cst_dwsrv_querymode 817
 u_dw 143
 pfc_PageSetup
 n_ds 493
 u_dw 144
 w_master 28
 pfc_PageSetupDlg
 n_ds 493
 u_dw 144
 pfc_Paste
 u_ddlb 121
 u_ddplb 128
 u_dw 144
 u_em 182
 u_mle 239
 u_oc 247
 u_rte 273
 u_sle 293
 pfc_PasteSpecial 247
 pfc_Populate
 u_lvs 212
 u_tvs 326
 pfc_PopulateDDDW 144
 pfc_PostOpen
 w_frame 14
 w_master 28
 pfc_PostUpdate
 n_ds 494
 u_base 366
 u_dw 145
 u_lvs 212
 u_tab 305
 u_tvs 326
 w_master 29
 pfc_PreAbout 564
 pfc_PreClose 29
 pfc_PreDebug 148
 pfc_PreDelete
 u_lvs 213
 u_tvs 327
 pfc_PreDeleteRow
 n_cst_dwsrv_linkage 735
 u_dw 146
 pfc_PreFindDlg 146
 pfc_PreInsert
 u_lvs 213
 u_tvs 327
 pfc_PreInsertRow
 n_cst_dwsrv_linkage 735
 u_dw 146
 pfc_PreLogonDlg 564
 pfc_PreMRUSave 30
 pfc_PreOpen 30
 pfc_PrePageSetupDlg
 n_ds 494
 u_dw 147
 pfc_PrePrintDlg
 n_ds 494
 u_dw 147
 u_rte 273
 pfc_PreRefreshItem
 u_lvs 214
 u_tvs 328
 pfc_PreRefreshLevel 328
 pfc_PreReplaceDlg 148
 pfc_PreRestoreRow
 u_dw 149
 pfc_PreRmbMenu
 u_ddlb 121
 u_ddplb 128
 u_dw 149
 u_em 182
 u_lvs 214
 u_mle 240
 u_oc 247
 u_rte 273
 u_sle 293
 u_tvs 328

pfc_PreSplash 565
pfc_PreToolbar 14
pfc_PreUpdate
 u_dw 150
 w_master 31
pfc_PrevDay 410
pfc_PreviousPage
 u_dw 150
 u_rte 274
pfc_PrevMonth 410
pfc_PrevWeek 411
pfc_Print
 n_ds 495
 u_dw 150
 u_rte 274
 w_master 31
pfc_PrintDlg
 n_ds 495
 u_dw 150
 u_rte 274
pfc_PrintImmediate
 n_ds 495
 u_dw 151
 u_rte 275
 w_frame 32
pfc_PrintPreview
 n_ds 496
 u_dw 151
 u_rte 275
pfc.Properties
 u_dw 151
 u_lvs 214
 u_tvs 329
pfc_RButtonDown
 n_cst_dwsrv_linkage 736
 n_cst_dwsrv_rowselection 909
pfc_RButtonUp 909
pfc_Refresh
 n_cst_lvsrv_datasource 1065
 u_lvs 214
pfc_RefreshItem
 n_cst_lvsrv_datasource 1065
 n_cst_tvsrv_levelsource 1296
 u_lvs 215
 u_tvs 329
pfc_RefreshLevel
 n_cst_tvsrv_levelsource 1297
 u_tvs 329
pfc_Rename
 u_lvs 215
 u_tvs 330
pfc_Replace
 n_cst_dwsrv_find 714
 n_cst_rtefind 1189
pfc_ReplaceAll
 n_cst_dwsrv_find 714
 n_cst_rtefind 1189
pfc_ReplaceDlg
 n_cst_dwsrv_find 714
 n_cst_rtefind 1190
 u_dw 152
 u_rte 275
pfc_ResetUpdate
 n_ds 496
 u_dw 152
pfc_Resize event
 n_cst_dwsrv_resize 880
 n_cst_resize 1180
 n_cst_winsrv_statusbar 1384
pfc_RestoreRow
 n_cst_dwsrv_rowmanager 896
 u_dw 152
pfc_Retrieve
 n_ds 496
 u_dw 152
 u_lvs 215
 u_tvs 330
pfc_RetrieveDDDW 153
pfc_RetrieveEnd 736
pfc_RetrieveStart 737
pfc_RowChanged
 n_cst_dwsrv_linkage 737
 u_dw 153
pfc_RowFocusChanged
 n_cst_dwsrv_linkage 737
 n_cst_dwsrv_rowselection 910
pfc_RowFocusChanging 738
pfc_RowValidation 153

pfc_Ruler
 n_ds 496
 u_dw 153
 u_rte 275
 pfc_Save
 u_rte 276
 w_master 32
 pfc_SaveAs
 u_rte 276
 w_master 33
 pfc_SaveObjects 33
 pfc_SearchCompare 331
 pfc_SelectAll
 u_ddlb 122
 u_ddplb 129
 u_dw 154
 u_em 183
 u_lb 201
 u_lvs 216
 u_mle 240
 u_plb 261
 u_rte 276
 u_sle 294
 pfc_SetItemAttributes
 n_cst_lvsrv_datasource 1066
 n_cst_tvsrv_levelsource 1297
 u_lvs 216
 u_tvs 331
 pfc_Show event 1385
 pfc_Sort 1104
 pfc_SortDlg
 n_cst_dwsrv_sort 923
 u_dw 154
 pfc_StatusBarClicked event 1385
 pfc_StatusBarDoubleClick event 1386
 pfc_StatusBarRButtonUp event 1386
 pfc_SystemError 566
 pfc_TileHorizontal
 n_cst_winsrv_sheetmanager 1374
 w_frame 14
 pfc_TileVertical
 n_cst_winsrv_sheetmanager 1374
 w_frame 14
 pfc_Timer
 n_cst_tmgtmultiple 1236, 1245
 pfc_Toolbars 15

pfc_Undo
 n_cst_lvsrv_datasource 1066
 n_cst_tvsrv_levelsource 1299
 u_dw 154
 u_em 183
 u_lvs 216
 u_mle 240
 u_rte 276
 u_sle 294
 u_tvs 332
 pfc_UndoArrange
 n_cst_winsrv_sheetmanager 1374
 w_frame 15
 pfc_UndoDelete
 n_cst_lvsrv_datasource 1067
 n_cst_tvsrv_levelsource 1299
 pfc_UndoEdit
 n_cst_lvsrv_datasource 1067
 n_cst_tvsrv_levelsource 1300
 pfc_UndoInsert
 n_cst_lvsrv_datasource 1068
 n_cst_tvsrv_levelsource 1300
 pfc_Update
 n_ds 497
 u_base 367
 u_dw 154
 u_lvs 217
 u_tab 305
 u_tvs 332
 w_master 34
 pfc_UpdateLinks 248
 pfc_UpdateObjects 35
 pfc_UpdatePrep
 n_ds 497
 u_base 367
 u_dw 155
 u_lvs 217
 u_tab 305
 u_tvs 332
 w_master 35
 pfc_UpdatesPending
 n_ds 497
 u_base 367
 u_dw 155
 u_lvs 217
 u_tab 306

u_tvs 333
w_master 35
pfc_UpdatesPendingRef 36
pfc_Validation
 n_ds 497
 u_base 368
 u_dw 155
 u_lvs 218
 u_tab 306
 u_tvs 333
 w_master 37
pfc_Values
 n_cst_dwsrv_querymode 817
 u_dw 155
pfc_Zoom
 n_ds 498
 u_dw 156
Picture 251
PictureBox 254
PictureHyperlink 257
PictureListBox 260
Pipeline object 527
platform service
 f_SetPlatform 79
 n_cst_platform 1158
 n_cst_platformwin32 1165
popup menu
 m_dw 60
 m_edit 62
 m_lvs 65
 m_oc 73
 m_tvs 75
preference service 1354
print preview service
 n_cst_dwsrv_printpreview 802
PrintHeader event 277
property service
 main discussion 811

Q

query mode service 815
queue object 1174

R

RadioButton 264
RButtonDown event
 u_dw 156
 u_progressbar 444
RButtonUp event
 u_ddlb 122
 u_ddplb 129
 u_dw 156
 u_em 183
 u_lvs 218
 u_mle 240
 u_oc 248
 u_rte 277
 u_sle 294
registry
 of_GetAppKey 568
 of_IsRegistryAvailable 572
 of_SetAppKey 574, 597
 of_SetUserKey 581, 601, 1139
report service
 n_cst_dssrv_report 650
 n_cst_dwsrv_report 826
required column 869
Resize event
 u_base 368
 u_dw 156
 u_tab 306
 w_frame 15
 w_master 37
resize service
 DataWindow 877
 n_cst_resize 1177
RetrieveEnd event 157
RetrieveStart event
 n_ds 498
 u_dw 157
RichTextEdit 267
RightClicked event
 u_lvs 218
 u_tvs 334
Rollback 544
row manager service 894
row selection services 904
RowFocusChanged event 157

S

s_pagesetupattrib 108
 s_paperattrib 109
 s_printdlgattrib 109
 s_svalue 110
 sample application 7
 security service
 n_cst_security 1193
 of_SetAppPreference 575
 of_SetMRU 579
 of_SetSecurity 579
 selection service 1197
 Service object 529
 sheet manager service 1372
 Show event 15
 SingleLineEdit 291
 Sort event 218
 sort service 920
 splash screen
 of_Splash 582
 pfc_PreSplash 565
 splitbar service 467
 SQL parsing service 1202
 SQL Spy
 main discussion 1205
 of_SetSQLSpy 640
 SQLPreview event
 n_ds 498
 u_dw 157
 StaticText 297
 StaticTextHyperlink 300
 status bar service 1379
 string parsing service 1214

T

Tab control 303
 TabPage 481
 token 1217
 transaction object 532
 transaction registration service
 main discussion 1272
 of_SetTrRegistration 580
 Transport object 548

TreeView 318
 TreeView base service 1281
 TreeView level source service 1290
 TreeView print service 1347

U

u_base 364
 u_calculator 380
 u_calendar 403
 u_cb 112
 u_cbx 115
 u_ddlb 118
 u_ddplb 125
 u_dw
 m_dw 60
 main discussion 132
 u_em 179
 u_gb 187
 u_gr 188
 u_hpb 191
 u_hsb 194
 u_htb 197
 u_lb 200
 u_lvs
 m_lvs 65
 main discussion 204
 u_mle 237
 u_oc
 m_oc 73
 main discussion 243
 u_p 251
 u_pb 254
 u_phl 257
 u_plb 260
 u_progressbar 441
 u_rb 264
 u_rte
 main discussion 267
 n_cst_rtefind 1187
 u_sle 291
 u_st 297
 u_st_splitbar 467
 u_sth 300

Index

u_tab
 main discussion 303
 n_cst_resize 1177
u_tabpg 481
u_tvs
 m_tvs 75
 main discussion 318
u_vpb 353
u_vsb 356
u_vtb 359
user INI file 581, 600, 1138
pfc_Zoom (n_ds) 498
pfc_Zoom (u_dw) 156

V

VerticalProgressBar 353
VerticalScrollBar 356
VerticalTrackBar 359

W

w_child 10
w_debuglog 639
w_frame 11
w_main 17
w_master
 main discussion 18
 n_cst_resize 1177
w_popup 53
w_response 54
w_sheet 57
w_sqlspy 1207
window services
 n_cst_winsrv 1351
 n_cst_winsrv_preference 1354
 n_cst_winsrv_sheetmanager 1372
 n_cst_winsrv_statusbar 1379

Z

zoom
 of_GetZoom 806
 of_SetRelativeZoom 866