POWERSOFT
ENTERPRISE
SERIES

POWERBUILDER LIBRARY FOR NETWARE

User's Guide

VERSION 4.0

PowerBuilder

November 1994

Contents

About This Manual

Subject

This manual describes the PowerBuilder Library for NetWare. Software with which you can quickly and easily develop PowerBuilder applications that integrate NetWare management procedures.

Organization

The first chapter gives instructions and information for getting started using the PowerBuilder Library for NetWare.

The remaining chapters describe the windows, DataWindow objects, functions, and external function calls used in the PowerBuilder Library for NetWare.

Software required

The PowerBuilder Libary for NetWare requires Version 4.0 of PowerBuilder, Novell NetWare Version 4.01 (or higher) running VLM Version 1.02, NWCALLS.DLL, NWNET.DLL, and NWPSRV.DLL.

Audience

This manual is for PowerBuilder users who want to create network-aware PowerBuilder applications. It assumes you are familiar with Novell NetWare and with the PowerBuilder development environment.

CHAPTER 1

Getting Started

About this chapter

This chapter provides step-by-step instructions for using the PowerBuilder Library for NetWare.

Contents

Before you begin

Make sure the subheading *NetWare DOS Requester* in your NET.CFG file contains this line:

PREFERRED TREE = (*your tree name here*)

About the PBNOVEL.PBL library

The objects, structures, and external function calls needed to build PowerBuilder applications that communicate with NetWare Version 4.01 running VLM Version 1.02 are in the PBNOVELL.PBL library.

This library is installed with the PowerBuilder Library for NetWare.

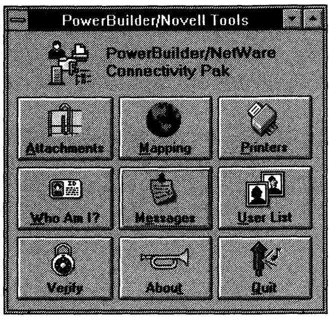
Understanding the sample application

The sample application installed with the PowerBuilder Library for NetWare provides a graphical way of maintaining network resources. These resources are grouped by commonly used NetWare functionality.

Functionality	Description
Attachments	Allows the user to attach and detach from network resources, view information on individual file servers, and change login passwords
Mapping	Allows the user to connect and disconnect drive mappings, view effective rights for a selected drive and view directories in the volumes of the attached file servers
Printers	Allows the user to connect and disconnect a print queue to an lpt device, view/modify printer options for a selected lpt device, and view the contents of a selected print queue
Who Am I	Displays who the user is in directory services and connection information on who the user is on all attached servers
Messages	Allows the user to send messages to other users and user groups across the network
User List	Displays a list of users currently logged in to a selected file server by name or connection number. Messages can be sent to users selected from the list
Verify Password	Verifies a user's password on the default file server

Main menu

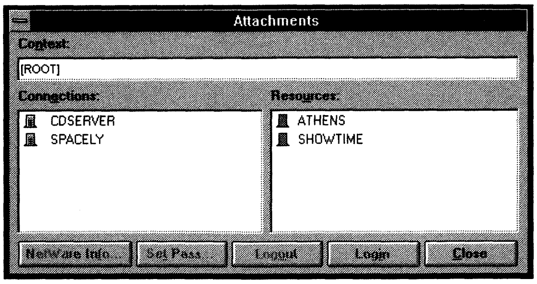
The purpose of the main menu of the sample application is to access the functionality of the PowerBuilder Library for NetWare.



To access a function, the user clicks the appropriate button or presses the associated quick-key (the underlined letter in the button name).

Attachments

The purpose of Attachments is to attach and detach from network resources, view information on individual file servers, and change login passwords.



[ROOT]

CDSERVER
SPACELY

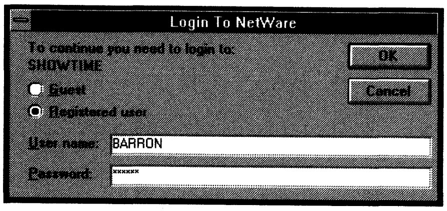
ATHENS
SHOWTIME

Field	Description
Context	Allows the user to set directory context for logging into a directory service tree or setting a password
Connections	Displays the user's current directory and bindery connections
Resources	Displays the resources available to the user

Field	Description
NetWare Info	Displays information on the selected NetWare server in the Connections list
Set Pass	Opens the Sct Password window allowing the user to change their password for the selected NetWare server in the Connections list
Logout	Detaches the user from the selected bindery server in the connections list. If a directory server was selected, detaches the user out of all directory server connections
Login	Opens the Login window allowing the user to attach to the selected resource in the Resource list
Close	Closes the attachments window

Login To NetWare

The purpose of Login To NetWare is to attach to the resource selected in the Resource list in the Attachments window. It logs in without running a login script.



To continue you need to login to:
SHOWTIME

○ Guest
● Registered user

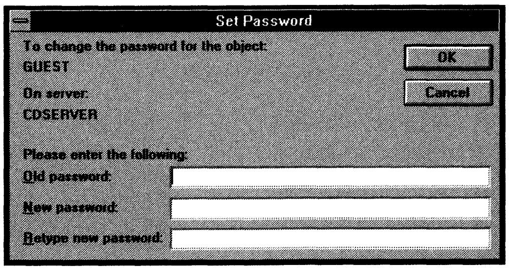
User name: BARRON
Password: xxxxxx

Field	Description
Guest	Fills the User Name field with *Guest*
Registered User	Fills the User Name field with the default user name if one exists
User Name	The user fills in the user name to use to log into the resource
Password	The user fills in the password associated with the name in the User Name field
OK	Accepts the User Name and Password and attempts to log in to the resource. If the login succeeds, closes the Login To NetWare window

Field	Description
Cancel	Cancels the login process and closes the Login To NetWare window

Set Password

The purpose of Set Password is to change the user's password for the resource selected in the Connections list on the Attachments window.


Set Password

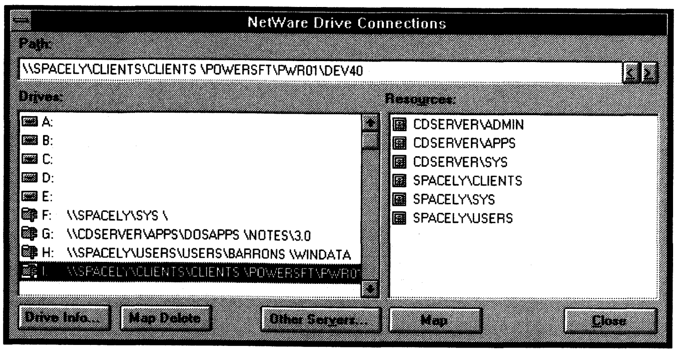
To change the password for the object:
GUEST
On server:
CDSERVER

Please enter the following:
Old password:
New password:
Retype new password:

Field	Description
Old Password	The user types the old password for the selected resource
New Password	The user types the desired new password
Retype New Password	The user types the new password again to confirm the change
OK	Accepts the new password. If new password is accepted, closes the Set Password window
Cancel	Cancels the new password and closes the Set Password window

Mapping

The purpose of Mapping is to connect and disconnect drive mappings, view effective rights for a selected drive, and view directories in the volumes of the attached file servers.

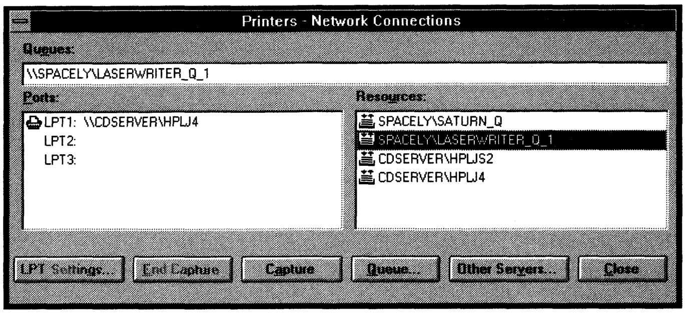


```
\\SPACELY\CLIENTS\CLIENTS \POWERSFT\PWR01\DEV40

A:
B:
C:
D:
E:
F:   \\SPACELY\SYS \
G:   \\CDSERVER\APPS\DOSAPPS \NOTES\3.0
H:   \\SPACELY\USERS\USERS\BARRONS \WINDATA
I:   \\SPACELY\CLIENTS\CLIENTS \POWERSFT\PWR01

CDSERVER\ADMIN
CDSERVER\APPS
CDSERVER\SYS
SPACELY\CLIENTS
SPACELY\SYS
SPACELY\USERS
```

Field	Description
Path	Indicates the volume and directory path for a drive
<	Moves the root of the path one directory to the left
>	Moves the root of the path one directory to the right
Drives	Displays a list of drive letters and their associated mappings (root and relative paths)
Resources	Displays a list of available directory map resources the user's workstation can map to
Drive Info	Displays the effective rights for the selected drive in the Drives list
Map Delete	Removes the mapping to the selected drive in the Drives list
Other Servers	Opens the Attachments window to allow the user to connect to and disconnect from other resources, and then refreshes the Resource list on the NetWare Drive Connections window
Map	Maps the path in the Path field to the selected drive in the Drives list
Close	Closes the NetWare Drive Connections window

Printers

The purpose of Printers is to connect and disconnect a print queue to an lpt device, view and modify printer options for a selected lpt device, and view the contents of a selected print queue.



\\SPACELY\LASERWRITER_Q_1

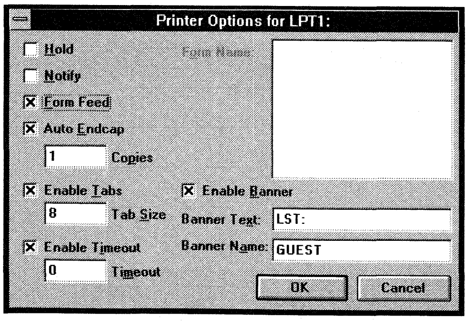
LPT1: \\CDSERVER\HPLJ4
LPT2:
LPT3:

SPACELY\SATURN_Q
SPACELY\LASERWRITER_Q_1
CDSERVER\HPLJS2
CDSERVER\HPLJ4

Field	Description
Queues	Indicates the queue to capture
Ports	Displays a list of available ports and the queues that they are currently capturing to
Resources	Displays a list of available print queue resources
LPT Settings	Opens the Printer Options windows for the selected lpt device in the Ports list
End Capture	Disconnects a print queue from the selected lpt device in the Ports list
Capture	Connects the print queue in the Queue field to the selected lpt device in the Ports list
Queue	Opens the Queue window allowing the user to view the print jobs waiting in the queue indicated in the Queues field
Other Servers	Opens the Attachments window allowing the user the connect to other servers, and then refreshes the Resource list
Close	Closes the Printers window

Printer Options

The purpose of Printer Options is to change capture options for the selected lpt device in the Ports list on the Printers window.

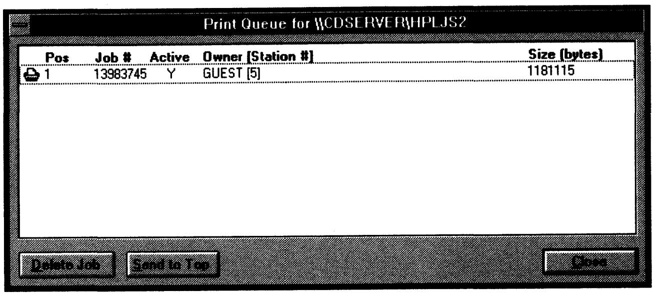


Hold: [ ]
Notify: [ ]
Form Feed: [X]
Auto Endcap: [X]
Copies: 1
Enable Tabs: [X]
Tab Size: 8
Enable Timeout: [X]
Timeout: 0
Form Name:
Enable Banner: [X]
Banner Text: LST:
Banner Name: GUEST

Field	Description
Hold	Holds all print jobs to the lpt device until this flag is turned off
Notify	*Notify field not working; unable to retrieve forms for bindery services*
Form Feed	Sends a form feed at the end of each print job
Auto Endcap	Specifies that the captured printing jobs be closed
Copies	Indicates the number of copies to be printed of each print job
Enable Tabs	Enables tabs of the size specified in Tab Size
Tab Size	Indicates the size of the tabs. Has effect only if Enable Tab is checked
Enable Timeout	Enables a timeout interval before the print buffer is closed and the job in the buffer is sent to the printer
Timeout	Indicates the amount of the timeout interval. Is valid only if Enable Timeout is checked
Form Name	Allows the user to select a defined print form (not enabled for bindery services)
Enable Banner	Enables a banner page to be printed at the start of the print job
Banner Text	Specifies the text to be printed on the banner page
Banner Name	Specifies the name to be printed on the banner page

Queue

The purpose of Queue is to display the contents of the print queue selected on the Printers window and control selected jobs (users can delete their own jobs from the queue or push a selected job to the top of the queue).



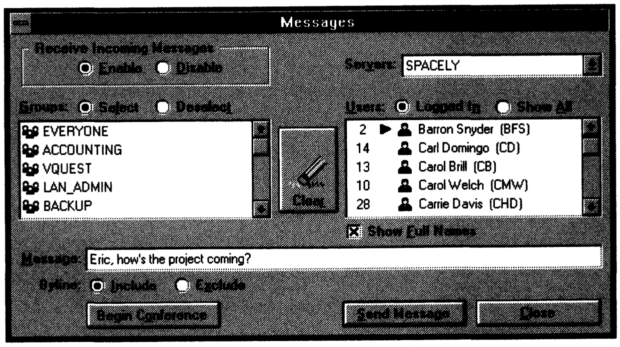
Print Queue for \\CDSERVER\HPLJS2

Pos	Job #	Active	Owner [Station #]	Size (bytes)
1	13983745	Y	GUEST [5]	1181115

Field	Description
Queue	Lists the print jobs in the selected queue
Delete Job	Removes the select print job from the queue
Send to Top	Sets the position of the selected print job in the queue to 1
Close	Closes the Queue window

Messages

The purposes of Messages is to send messages and conference with selected users.



Messages

Receive Incoming Messages: [✓ Enable] Disable

Servers: SPACELY

Groups: [✓ Select] Deselect

EVERYONE
ACCOUNTING
VQUEST
LAN_ADMIN
BACKUP

Users: [✓ Logged In] Show All

2	▶ Barron Snyder (BFS)
14	Carl Domingo (CD)
13	Carol Brill (CB)
10	Carol Welch (CMW)
28	Carrie Davis (CHD)

[✓ Show Full Names]

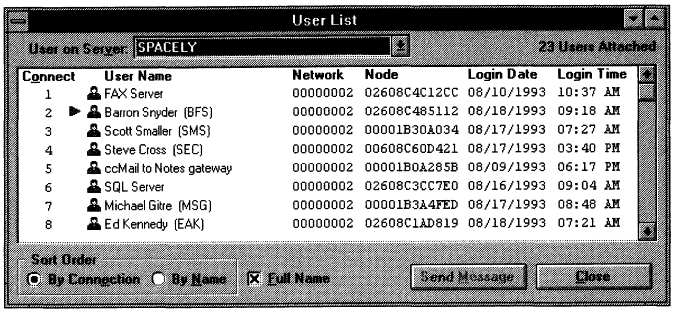
Message: Eric, how's the project coming?

Byline: [✓ Include] Exclude

Field	Description
Enable/Disable Incoming Messages	Allows or prevents the workstation to receive broadcast messages
Servers	Lists the servers that are currently attached
Groups: Select/Deselect	Selects (highlights) or deselects a group and its members
Groups	Lists the groups on the currently selected server. The user clicks a group to send messages to its members
Users: Logged In/Show All	Shows all users listed in a server or just the users currently logged in
Users	Lists the users and their connection numbers. Click the users you want to send messages to
Show Full Names	Shows the full name of each user if the server supports this. Otherwise, the login names display
Clear	Clears the highlights from the group data window and the user data window, and clears the Message field
Message	Contains user-entered message text
Byline: Include/Exclude	Includes or excludes the sender's name and connection number from the message
Begin/End Conference	Begins or ends conference mode (the window expands to include the area that displays messages from all conference participants and automatically scrolls as new messages are added)
Send Message	Sends message text to highlighted names in user list
Close	Closes the Messages window

User List

The purpose of User List is to display a list of users currently logged in to a selected file server by name or connection number. Messages can be sent to users selected from the list.



User on Server: SPACELY — 23 Users Attached

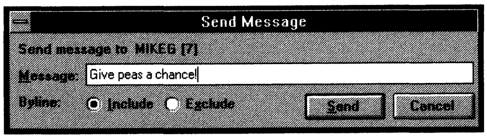
Connect		User Name	Network	Node	Login Date	Login Time
1		FAX Server	00000002	02608C4C12CC	08/10/1993	10:37 AM
2	▶	Barron Snyder (BFS)	00000002	02608C485112	08/18/1993	09:18 AM
3		Scott Smaller (SMS)	00000002	00001B30A034	08/17/1993	07:27 AM
4		Steve Cross (SEC)	00000002	00608C60D421	08/17/1993	03:40 PM
5		ccMail to Notes gateway	00000002	00001B0A285B	08/09/1993	06:17 PM
6		SQL Server	00000002	02608C3CC7E0	08/16/1993	09:04 AM
7		Michael Gitre (MSG)	00000002	00001B3A4FED	08/17/1993	08:48 AM
8		Ed Kennedy (EAK)	00000002	02608C1AD819	08/18/1993	07:21 AM

Sort Order: [✓ By Connection] By Name — [✓ Full Name]

Field	Description
User on Server	A dropdown listbox containing the names of all connected file servers. Selecting a different file server changes the user list
Users Attached	Indicates how many user are attached to the selected file server
User List	Displays a list of users currently attached to the selected server. A red arrow identifies the current user
Sort Order	Sorts the user list by connection number or name
Full Name	Displays users' full names in the connection list if the server supports this. Otherwise, the login names display
Send Message	Opens the Send Message window so the user can send a message to the currently selected user in the User List
Close	Closes the User List window

Send Message

The purpose of Send Message is to send a message to another user.



Send message to MIKEG [7]

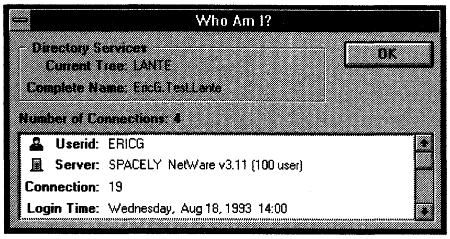
Message: Give peas a chance!

Byline: [✓ Include] Exclude

Field	Description
Message	Contains the message to be sent to the selected user
Byline	Includes or excludes the sending user's name from the message
Send	Sends the message to the user and closes the Send Message window
Cancel	Cancels the message and closes the Send Message window

Who Am I

The purpose of Who Am I is to display the complete name of a user in Directory Services and connection information for the user on all attached servers.

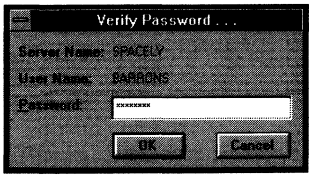


Directory Services
Current Tree: LANTE
Complete Name: EricG.Test.Lante
Number of Connections: 4
Userid: ERICG
Server: SPACELY NetWare v3.11 (100 user)
Connection: 19
Login Time: Wednesday, Aug 18, 1993 14:00

Field	Description
Current Tree	Displays the user's current directory tree
Complete Name	Displays the user's complete directory services name
Number of Connections	Displays the current number of connections the user has to file servers
Connection List	Displays the list of information for each connection the user has to a file server

Verify Password

The purpose of Verify Password is to verify a user's password on the default file server.



Server Name: SPACELY
User Name: BARRONS
Password: xxxxxxxx

Field	Description
Password	Displays the user's password for the default server
OK	Verifies the user's password and closes the Verify Password window
Cancel	Cancels the verification and closes the Verify Password window

CHAPTER 2

Window Information

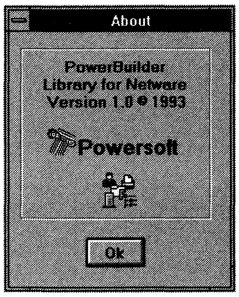
About this chapter

This chapter describes each of the windows used in the PowerBuilder Library for NetWare. Each description lists the controls, events, and functions used in the window.

Contents

The windows are listed alphabetically.

w_about



Description Displays general information for the application.

Parameters None

Controls used

Control name	Type
cb_ok	CommandButton
p_lante_logo	Picture
p_pbnovell	Picture
p_powersoft_logo	Picture
st_and	StaticText
st_app_name	StaticText
st_date	StaticText
st_version	StaticText

Window events None

Window functions None

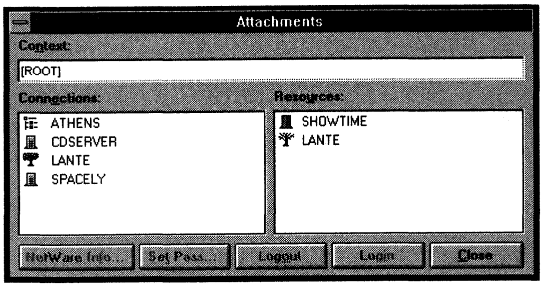
Control event

Event	Description
cb_ok.clicked	Close the About window

Required external functions

None

w_attachments



Description

Allows the user to view individual server information, change login passwords, attach network connections, and detach network connections.

Parameters

None

Controls used

Control name	Type
cb_close	CommandButton
cb_login	CommandButton
cb_logout	CommandButton
cb_netware_info	CommandButton
cb_set_pass	CommandButton
dw_attached	DataWindow
dw_unattached	DataWindow
sle_context	SingleLineEdit
st_connections	StaticText
st_context	StaticText
st_resources	StaticText

Window events

Event	Description
open	Retrieves list of connected servers and unattached resources and displays them in the w_attachments window

Window functions

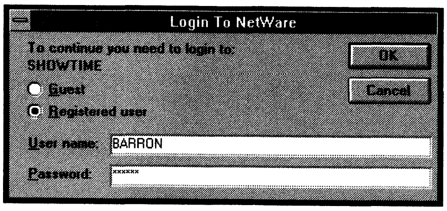
Function	Use to
wf_get_attached_servers	Retrieve a list of attached servers and bindery service objects, indicate the primary server, and display them on the window
wf_get_unattached_servers	Retrieve a list of all available resources and display them in the dw_unattached DataWindow

Control events

Event	Description
cb_set_pass.clicked	Opens the set password window for the selected server or directory service object
cb_logout.clicked	Logs the user out of the selected server or out of all directory services objects
cb_login.clicked	Logs the user into the selected server or into all directory services objects by opening the w_login window
cb_netware_info.clicked	Displays NetWare information for the selected server or directory service object
dw_attached.clicked	Highlights the clicked row and enable/disable buttons based on the value in the clicked row
dw_attached.rowfocuschanged	Highlights row and enable/disable buttons based on selected server or directory service object
dw_attached.doubleclicked	Displays Netware information for the selected server or Directory Service object that was double-clicked
dw_unattached.rowfocuschanged	Highlights the selected row and enable/disable buttons based on selection
dw_unattached.clicked	Highlights the clicked row and enable/disable buttons based on selection
dw_unattached.doubleclicked	Triggers the login button for the double-clicked row
cb_close.clicked	Closes the w_attachments window

Required external functions	NWGetDefaultConnectionID NWGetPreferredConnName NWScanObject

w_login



To continue you need to login to:
SHOWTIME

Guest [✓ Registered user]

User name: BARRON

Password: ******

Description

Allows the user to log in to bindery services or directory services based on the server name and type

Parameters

None

Controls used

Control name	Type
cb_cancel	CommandButton
cb_ok	CommandButton
dw_login	DataWindow
gb_login	GroupBox
rb_guest	RadioButton
rb_registered_user	RadioButton
st_continue	StaticText
st_server_name	StaticText

Window event

Event	Description
Open	Retrieves user name of primary network connection and sets it as the default user name

Window functions

None

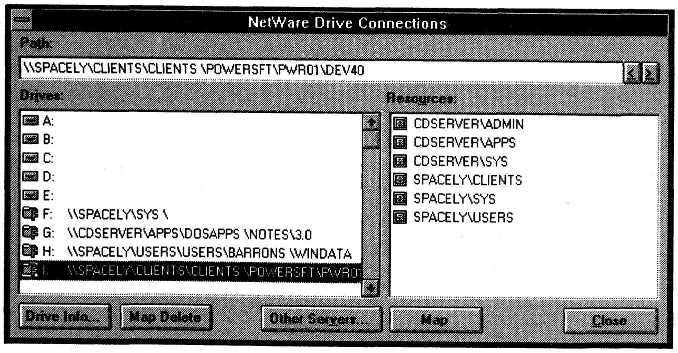
Control events

Event	Description
rb_guest.clicked	Sets user name field in DataWindow based on radio button selection. If checked, puts GUEST in the user name field; otherwise, puts the default user name selection
cb_cancel.clicked	Cancels the login process by closing the parent window
cb_ok.clicked	Logs into directory services or a bindery server based on the directory services flag and the parameter passed into the window

Required external functions

NWGetConnectionStatus
NWGetDefaultConnectionID

w_map



Path: \\SPACELY\CLIENTS\CLIENTS \POWERSFT\PWR01\DEV40

Drives:

```
A:
B:
C:
D:
E:
F:   \\SPACELY\SYS \
G:   \\CDSERVER\APPS\DOSAPPS \NOTES\3.0
H:   \\SPACELY\USERS\USERS\BARRONS \WINDATA
I:   \\SPACELY\CLIENTS\CLIENTS \POWERSFT\PWR01
```

Resources:

```
CDSERVER\ADMIN
CDSERVER\APPS
CDSERVER\SYS
SPACELY\CLIENTS
SPACELY\SYS
SPACELY\USERS
```

Description

Allows users to view their effective rights on a drive and connect and disconnect drive mappings

Parameters

None

Controls used

Control name	Type
cb_close	CommandButton
cb_drive_info	CommandButton
cb_left	CommandButton
cb_map	CommandButton
cb_map_delete	CommandButton
cb_other_servers	CommandButton
cb_right	CommandButton
dw_data_drives	DataWindow
dw_server_volume	DataWindow
sle_map_path	SingleLineEdit
st_drives	StaticText
st_path	StaticText
st_resources	StaticText

Window events

Event	Description
Open	Retrieves the drive mapping list and a list of servers and their available volumes

Window functions

Function	Use to
wf_get_drives	Retrieve a list of drive mappings and display them in the dw_data_drives DataWindow
wf_get_server_volumes	Retrieve a list of the server volumes foR all available servers to be imported into the dw_server_volume DataWindow

Control events

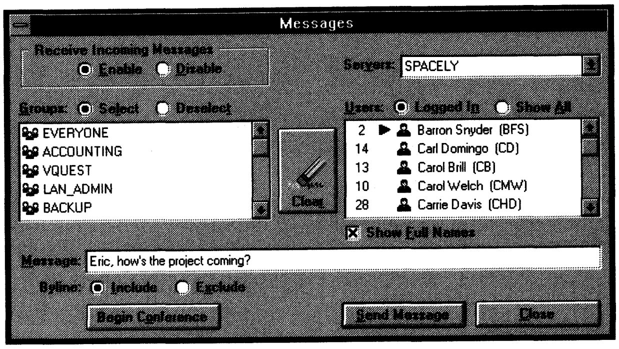
Event	Description
cb_other_servers.clicked	Opens w_attachments to allow users to log in/log off other servers bindery objects. Retrieves new volumes when user returns
cb_right.clicked	Moves the root path one directory to the right
cb_left.clicked	Moves the root path one directory to the left
cb_map.clicked	Maps the entered path to the selected drive
cb_map_delete.clicked	Deletes the path associated with the selected drive
cb_drive_info.clicked	Retrieves and displays NetWare drive information for the selected drive
dw_data_drives.clicked	Highlights selected row and enable/disable buttons based on selected row
dw_data_drives.rowfocuschanged	Highlights selected row and enable/disable buttons based on selected row
dw_data_drives.doubleclicked	Retrieves drive information for the double-clicked row

Event	Description
cb_close.clicked	Closes the w_map window
dw_server_volume.clicked	Highlights selected row and enable/disable buttons based on selected row
dw_server_volume.rowfocuschanged	Highlights selected row and enable/disable buttons based on selected row
dw_server_volume.doubleclicked	Displays subdirectories of the volume/path that has been double-clicked. If the volume/path select already has subdirectories displayed, removes the subdirectories

Required external functions

NWDeleteDriveBase
NWParseNetWarePath
NWScanDirectoryInformation2

w_messages



Receive Incoming Messages: [✓ Enable] Disable

Servers: SPACELY

Groups: [✓ Select] Deselect

- EVERYONE
- ACCOUNTING
- VQUEST
- LAN_ADMIN
- BACKUP

Users: [✓ Logged In] Show All

```
 2  ▶ Barron Snyder  (BFS)
14    Carl Domingo  (CD)
13    Carol Brill  (CB)
10    Carol Welch  (CMW)
28    Carrie Davis  (CHD)
```

[✓ Show Full Names]

Message: Eric, how's the project coming?

Byline: [✓ Include] Exclude

Description

Allows the user to send messages to other NetWare users, enable/disable broadcast messaging, and have online conferences

Parameters

None

Controls used

Control name	Type
cb_close	CommandButton
cb_conference	CommandButton
cb_send_message	CommandButton
cbx_full_names	CheckBox
ddlb_servers	DropDownListBox
dw_groups	DataWindow
dw_users	DataWindow
gb_byline	GroupBox
gb_groups	GroupBox
gb_receive	GroupBox
gb_users	GroupBox
lb_chatter	ListBox
pb_clear	PictureButton
rb_deselect	RadioButton
rb_disable_messages	RadioButton

Control name	Type
rb_enable_messages	RadioButton
rb_exclude_byline	RadioButton
rb_include_byline	RadioButton
rb_logged_in	RadioButton
rb_select	RadioButton
rb_show_all	RadioButton
sle_message	SingleLineEdit
st_byline	StaticText
st_groups	StaticText
st_message	StaticText
st_servers	StaticText
st_users	StaticText

Window events

Event	Description
Open	Retrieves connected server names for DropDownListBox, displays default server, and calls window functions to fill the user list DataWindow and the groups DataWindow
Timer	Retrieves messages and adds them to ListBox

Window functions

Function	Use to
wf_get_all_users	Retrieve a list of users from the selected file server and populate a DataWindow (dw_users) with the information
wf_get_default_server_name	Retrieve the default server name
wf_get_group	Retrieve list of groups to populate DataWindow (dw_groups)
wf_highlight_group_members	Highlight members (in dw_users) of selected group
wf_send_message	Send broadcast message to all selected recipients

Function	Use to
wf_update_user_list	Retrieve a list of logged-in users from the selected file server and populate a DataWindow (dw_users) with the information

Control events

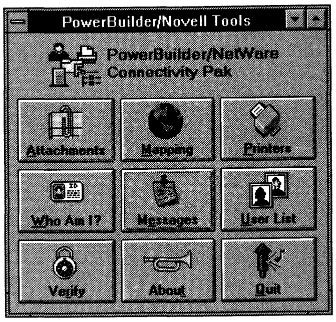
Event	Description
cb_conference.clicked	Turns on (or off) conferencing
ddlb_servers.selectionchanged	Calls functions to update user list and groups according to which server is selected
cb_close.clicked	Closes Messages window
cb_send_message.clicked	Sends message
cbx_full_names.clicked	Displays full names of users (if available)
pb_clear.clicked	Clears highlighted users and groups and clears message text
dw_users.clicked	Highlights (or unhighlight if already highlighted) clicked row
dw_groups.clicked	Highlights members of selected group
rb_logged_in.clicked	Updates DataWindow with logged-in users
rb_show_all.clicked	Updates DataWindow with all users (user objects on server)
rb_disable_messages.clicked	Disallows broadcast messages to be received
rb_enable_messages.clicked	Allows broadcast messages to be received

Required external functions

NWDisableBroadcasts
NWEnableBroadcasts
NWGetBroadcastMessage
NWGetConnectionHandle
NWGetConnectionInformation
NWGetConnectionNumber
NWGetDefaultConnectionID
NWGetFileServerInformation
NWGetFileServerName
NWGetObjectConnectionNumbers

NWIsObjectInSet
NWReadPropertyValue
NWScanObject
NWSendBroadcastMessage
NWSetBroadcastMode

w_pbnovell_tools



Description

The main menu of the sample application, which demonstrates the PowerBuilder Novell Connectivity Pak

Parameters

None

Controls used

Control name	Type
pb_novell	Picture
pb_about	PictureButton
pb_attachments	PictureButton
pb_exit	PictureButton
pb_mapping	PictureButton
pb_messages	PictureButton
pb_password	PictureButton
pb_printer	PictureButton
pb_user_list	PictureButton
pb_who_am_i	PictureButton
st_name	StaticText

Window event

Event	Description
Open	Ensures that the workstation is running the correct version of NetWare and displays icons for the various windows of the application

Window functions

None

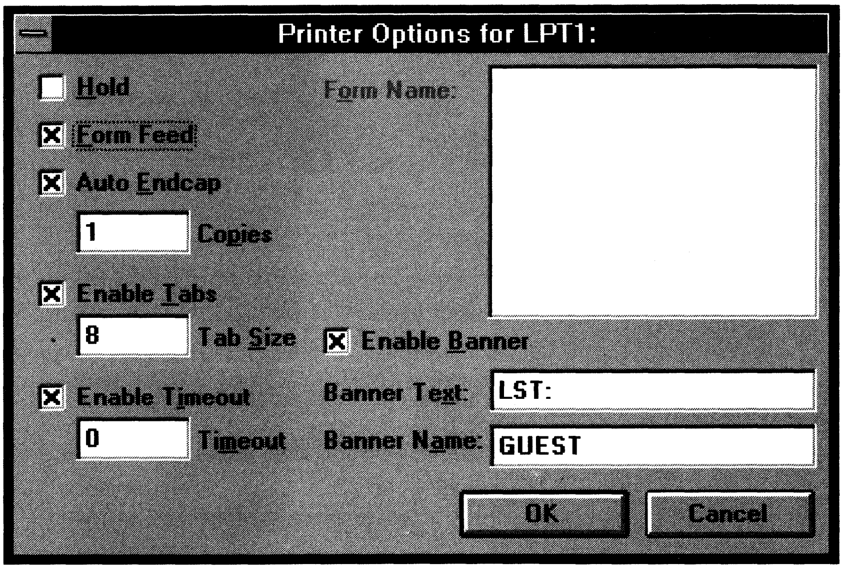
Control events

Event	Description
pb_password.clicked	Opens the verify password window
pb_about.clicked	Opens w_about
pb_who_am_i .clicked	Opens w_who_am_i
pb_messages.clicked	Opens w_messages
pb_mapping.clicked	Opens w_map
pb_user_list.clicked	Opens w_user_list
pb_exit.clicked	Quits the application
pb_printer.clicked	Opens w_printers
pb_attachments.clicked	Opens w_attachments

Required external functions

None

w_printer_options


Printer Options for LPT1:

- [ ] Hold
- [x] Form Feed
- [x] Auto Endcap

Copies: 1

- [x] Enable Tabs

Tab Size: 8

- [x] Enable Timeout

Timeout: 0

Form Name:

- [x] Enable Banner

Banner Text: LST:

Banner Name: GUEST

Description Allows the user to set the printer options for the selected lpt device

Parameters None

Controls used

Control name	Type
cb_cancel	CommandButton
cb_ok	CommandButton
cbx_auro_endcap	CheckBox
cbx_enable_banner	CheckBox
cbx_enable_tabs	CheckBox
cbx_enable_timeout	CheckBox
cbx_form_feed	CheckBox
cbx_hold	CheckBox
cbx_notify	CheckBox
lb_form_name	ListBox

Control name	Type
sle_banner_name	SingleLineEdit
sle_banner_text	SingleLineEdit
sle_copies	SingleLineEdit
sle_tab_size	SingleLineEdit
sle_timeout	SingleLineEdit
st_bannername	StaticText
st_bannertext	StaticText
st_copies	StaticText
st_form_name	StaticText
st_tabsize	StaticText
st_timeout	StaticText

Window event

Event	Description
Open	Retrieves the printer options available for the seleted lpt device

Window functions

None

Control events

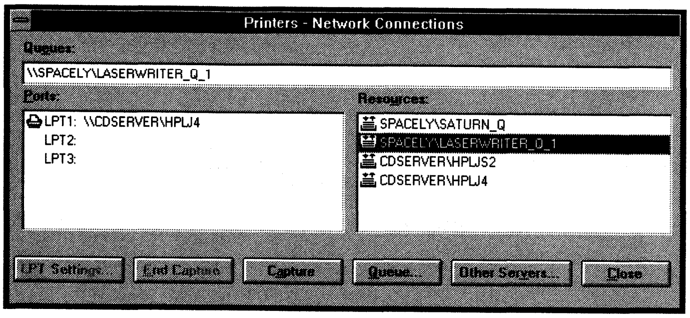
Event	Description
sle_banner_name.modified	Sets banner name length limits
sle_banner_text. modified	Sets banner text length limits
sle_timeout.modified	Sets Timeout limits
sle_tab_size.modified	Sets Tab size limits
sle_copies.modified	Sets number of copies limit
cbx_enable_banner.clicked	Sets the enable banner flag and enables the banner name and banner text fields if the cbx_enable_banner has been checked
cbx_enable_timeout.clicked	Enables the sle_timeout field if cbx_enable_timeout has been checked; otherwise, disables the field

Event	Description
cbx_enable_tabs.clicked	Sets the enabled flag in the printer flags and enables sle_tab_size if enabled flag is checked; otherwise, disables sle_tab_size
cbx_auto_endcap.clicked	Sets the flushCaptureOnClose flag in the istrct_nwcapture_flags1 structure
cbx_form_feed.clicked	Sets the print flag for formfeed
cb_cancel.clicked	Closes the w_printer_options window
cb_ok.clicked	Sets the printer options for the select lpt device based on selections made in the w_printer_options window

Required structures and external functions

s_connect_info
NWDSCreateContext
NWDSFreeContext
NWDSSetContext
NWGetBannerUserName
NWGetCaptureFlags
NWGetConnectionStatus
NWIsDSServer
NWPSPdfScanForm
NWSetBannerUserName
NWSetCaptureFlags

w_printers



\\SPACELY\LASERWRITER_Q_1

LPT1: \\CDSERVER\HPLJ4
LPT2:
LPT3:

SPACELY\SATURN_Q
SPACELY\LASERWRITER_Q_1
CDSERVER\HPLJS2
CDSERVER\HPLJ4

Description

Allows the user to view and set printer options for a selected lpt device and connect and disconnect a print queue to lpt devices

Parameters

None

Controls used

Control name	Type
cb_close	CommandButton
cb_connect	CommandButton
cb_disconnect	CommandButton
cb_options	CommandButton
cb_other_servers	CommandButton
cb_queue	CommandButton
dw_lpt1_server_name	DataWindow
dw_server_queue_name	DataWindow
sle_server_queue	SingleLineEdit
st_ports	StaticText
st_queues	StaticText
st_resources	StaticText

Window event

Event	Description
Open	Retrieves a list of the current lpt device settings and a list of available print queues and displays them in w_printers

Window functions

Function	Use to
w_get_lpt_list	Retrieve a list a lpt devices and their associated settings
w_get_queue_list	Retrieve a list of all available printing queues and display on w_printers

Control event

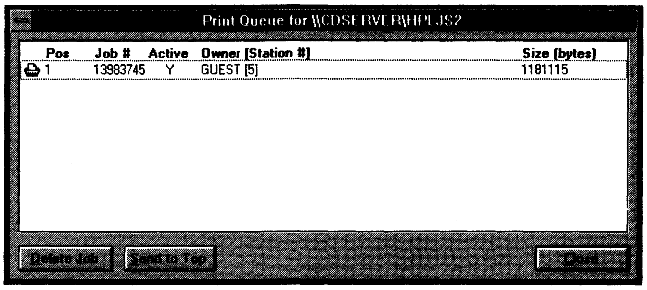
Event	Description
cb_queue.clicked	Displays the contents of the selected queue in the w_queue window
dw_lpt1_server_name.clicked	Enables/disables command buttons based on the status of the selected lpt device and to fill sle_server_queue with the selected server and queue
dw_lpt1_server_name.rowfocuschanged	Enables/disables command buttons based on the status of the selected lpt device and to fill sle_server_queue with the selected server and queue
dw_lpt1_server_name.doubleclicked	Displays the printer optins of the double-clicked lpt device
dw_server_queue_name.clicked	Enables/disables command buttons based on the status of the selected server and queue and to display the selected name in sle_server_queue
dw_server_queue_name.rowfocuschanged	Enables/disables command buttons based on the status of the selected server and queue and to display the selected name in sle_server_queue
dw_server_queue_name.doubleclicked	Displays the contents of the selected server and queue in the w_queue window.

Event	Description
cb_options.clicked	Allows the user to change the printer options of the specified lpt device in the w_printer_options window
cb_other_servers.clicked	Allows the user to change their network connections in the w_attachments window
cb_disconnect.clicked	Disconnects the print queue from the lpt device
cb_connect.clicked	Connects a print queue to a selected lpt device
cb_close.clicked	Closes the w_printers window

Required external functions

NWEndCapture
NWFlushCapture
NWStartQueueCapture

w_queue



Pos	Job #	Active	Owner [Station #]	Size (bytes)
1	13983745	Y	GUEST [5]	1181115

Description

Allows users to view the contents of a print queue and delete any of their own print jobs from the queue or move one of their own print jobs to the top of the queue

Parameters

None

Controls used

Control name	Type
cb_close	CommandButton
cb_delete_job	CommandButton
cb_send_to_top	CommandButton
dw_queue	DataWindow

Window events

Event	Description
Open	Displays contents of queue for supplied connection and queue
Timer	Displays contents of queue for supplied connection and queue

Window functions

None

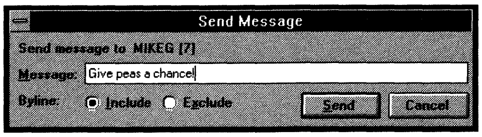
Control events

Event	Description
cb_send_to_top.clicked	Sends the selected print job to the top of the queue
cb_delete_job.clicked	Deletes the selected print job from the queue
cb_close.clicked	Closes the w_queue window
dw_queue.clicked	Highlights the selected row
dw_queue.rowfocuschanged	Highlights the selected row

Required external functions

NWChangeQueueJobPosition2
NWGetConnectionHandle
NWRemoveJobFromQueue2

w_send_message



Description

Allows the user to send a message to another NetWare user selected on the w_user_list window

Parameters

None

Controls used

Control name	Type
cb_cancel	CommandButton
cb_send	CommandButton
gb_byline	GroupBox
rb_exclude_byline	RadioButton
rb_include_byline	RadioButton
sle_message	SingleLineEdit
st_byline	StaticText
st_message	StaticText
st_send_message_to	StaticText
st_user_name	StaticText

Window event

Event	Description
Open	Sets up message sender's and recipient's names and IDs in order to send a message

Window function

Function	Use to
wf_send_message	Send a message to the selected user

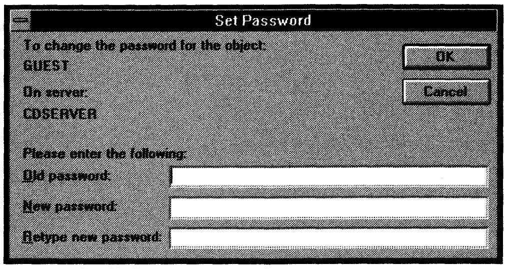
Control events

Event	Description
cb_cancel.clicked	Closes the w_send_message window
cb_send.clicked	Calls window function to send message

Required external functions

NWGetConnectionHandle
NWSendBroadcastMessage

w_set_password



To change the password for the object:
GUEST
On server:
CDSERVER

Please enter the following:
Old password:
New password:
Retype new password:

Description

Allows users to change their password for a bindery server or a directory services tree (this window is opened from w_attachments)

Parameters

None

Controls used

Control name	Type
cb_cancel	CommandButton
cb_ok	CommandButton
dw_password	DataWindow

Window event

Function	Use to
Open	Retrieve default user name for supplied server

Window functions

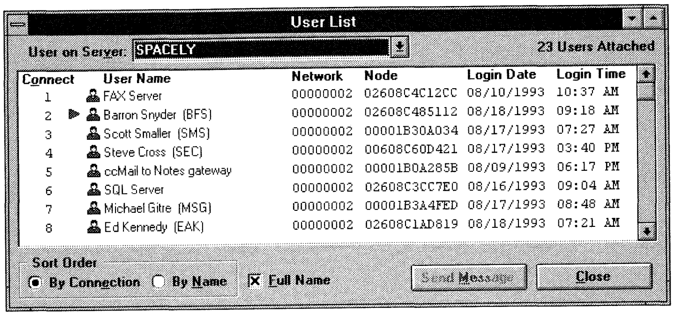
None

Control events

Event	Description
cb_cancel.clicked	Closes the w_set_password window
cb_ok.clicked	Accepts password changes

Required structures and external functions	s_connect_info NWGetConnectionHandle NWGetConnectionStatus

w_user_list



User on Server: SPACELY

23 Users Attached

Connect	User Name	Network	Node	Login Date	Login Time
1	FAX Server	00000002	02608C4C12CC	08/10/1993	10:37 AM
2	Barron Snyder (BFS)	00000002	02608C485112	08/18/1993	09:18 AM
3	Scott Smaller (SMS)	00000002	00001B30A034	08/17/1993	07:27 AM
4	Steve Cross (SEC)	00000002	00608C60D421	08/17/1993	03:40 PM
5	ccMail to Notes gateway	00000002	00001B0A285B	08/09/1993	06:17 PM
6	SQL Server	00000002	02608C3CC7E0	08/16/1993	09:04 AM
7	Michael Gitre (MSG)	00000002	00001B3A4FED	08/17/1993	08:48 AM
8	Ed Kennedy (EAK)	00000002	02608C1AD819	08/18/1993	07:21 AM

Sort Order: [✓ By Connection] By Name

[✓ Full Name]

Description

Lists all users currently logged into the selected server by name or connection number (from this list a user can select another NetWare user to send a message to)

Parameters

None

Controls used

Control name	Type
cb_close	CommandButton
cb_send_message	CommandButton
cbx_full_name	CheckBox
ddlb_servers	DropDownListBox
dw_user_list	DataWindow
gb_sort_order	GroupBox
rb_by_connection	RadioButton
rb_by_name	RadioButton
st_number_users	StaticText
st_server	StaticText

Window event

Event	Description
Open	Displays a list of users and corresponding information for the selected window

Window functions

None

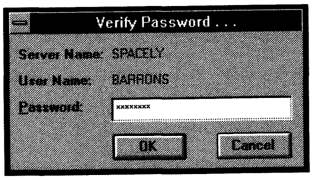
Control events

Event	Description
ddlb_servers.selectionchanged	Retrieves the list of users and corresponding information for the selected server
cb_close.clicked	Closes the w_user_list window
cb_send_message.clicked	Sends a message to the selected user
cbx_full_name.clicked	Displays either an object's full name or its short name based on status of this checkbox
rb_by_name.clicked	Sorts the DataWindow by user name (either object name or full name) and Connection ID
rb_by_connection.clicked	Sorts the DataWindow by Connection ID
dw_user_list.clicked	Highlights the clicked row
dw_user_list.rowfocuschanged	Highlights the selected row

Required external functions

NWGetConnectionHandle
NWGetConnectionNumber
NWGetDefaultConnectionID
NWGetFileServerName

w_verify_password



Description Allows the user to verify their password for the default server

Parameters None

Controls used

Control name	Type
cb_cancel	CommandButton
cb_ok	CommandButton
sle_password	SingleLineEdit
st_password	StaticText
st_server	StaticText
st_server_name	StaticText
st_user	StaticText
st_user_name	StaticText

Window event

Event	Description
open	Fills w_verify password with primary connection server and user name based on whether the user is logged into the bindery or directory services

Window functions None

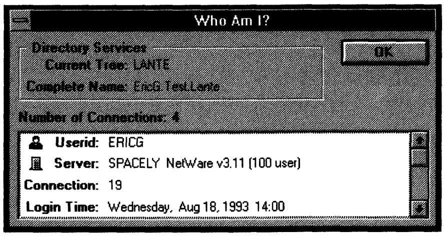
Control events

Event	Description
cb_cancel.clicked	Closes the w_verify window
cb_ok.clicked	Verifies the password

Required external functions

NWGetConnectionStatus
NWGetDefaultConnectionID
NWIsDSServer
NWVerifyObjectPassword

w_who_am_i



Directory Services
Current Tree: LANTE
Complete Name: EricG.Test.Lante
Number of Connections: 4

Userid: ERICG
Server: SPACELY NetWare v3.11 (100 user)
Connection: 19
Login Time: Wednesday, Aug 18, 1993 14:00

Description

Displays the user's current connection information for bindery and directory services

Parameters

None

Controls used

Control name	Type
cb_ok	CommandButton
dw_who_am_i	DataWindow
gb_directory_services	GroupBox
st_complete_name	StaticText
st_complete_name_label	StaticText
st_current_tree	StaticText
st_current_tree_label	StaticText
st_number_of_connections	StaticText

Window event

Event	Description
Open	Retrieves Who Am I information from directory services (if connected) and all connected servers

Window functions

None

Control event

Event	Description
cb_ok.clicked	Closes the w_who_am_i window

Required external functions

NWIsDsAuthenticated

CHAPTER 3

DataWindow Information

About this chapter

This chapter describes each DataWindow object used in the PowerBuilder Library for NetWare. The descriptions include the data type and name of each column and a picture of the DataWindow object.

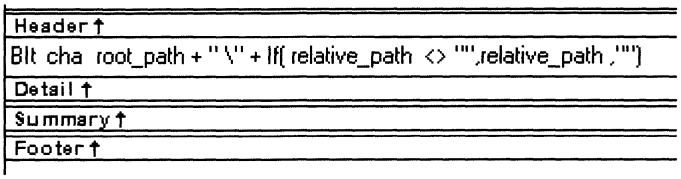
Contents

The DataWindow objects are listed alphabetically.

d_drive_paths

The DataWindow d_drive_paths is used on w_map with the Windows function wf_get_drives.

number	Drive_Number
number	Drive_Status
number	Connection_ID
char(304)	Root_Path
char(304)	Relative_Path
char(304)	Full_Path

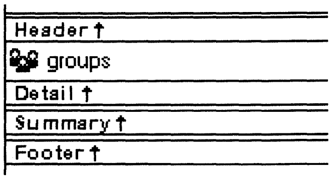


Header ↑
Blt cha root_path + " \" + If(relative_path <> "",relative_path ,"")
Detail ↑
Summary ↑
Footer ↑

d_groups

The DataWindow d_groups is used on w_messages with the Windows function wf_get_groups.

char(48)	Groups

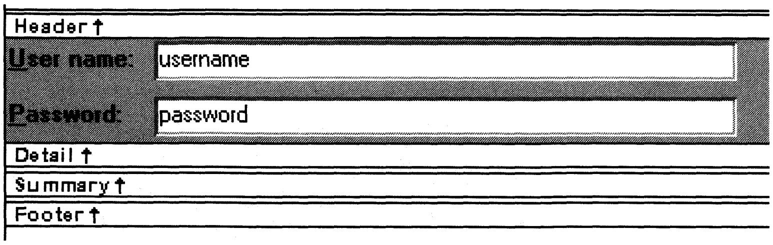


Header ↑
groups
Detail ↑
Summary ↑
Footer ↑

d_login

The DataWindow d_login is used on w_login.

char(256)	UserName
char(48)	Password
char(48)	UserNameDefault

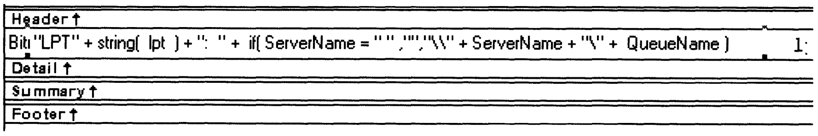


```
Header ↑
User name:  username
Password:   password
Detail ↑
Summary ↑
Footer ↑
```

d_lpt_server_queue_name

The DataWindow d_lpt_server_queue_name is used on w_printers with the function f_get_lpt_list.

number	lpt
char(48)	ServerName
number	ServerID
char(48)	QueueName
number	QueueID
number	PermanentFlag

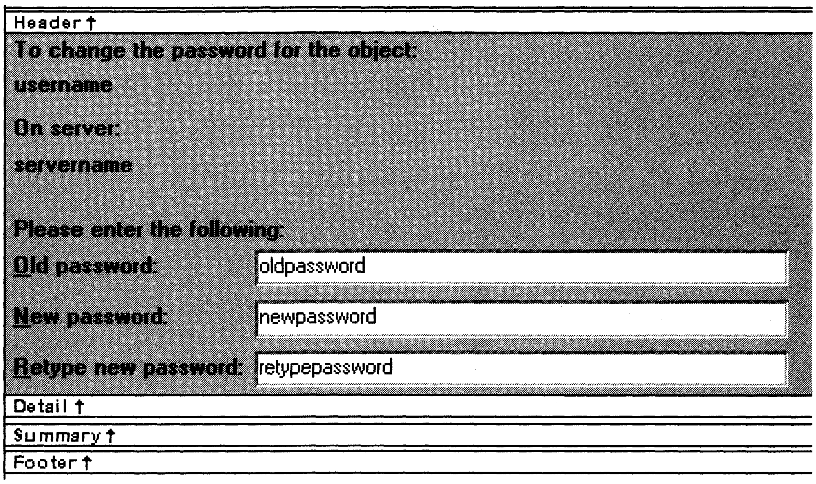


```
Header ↑
Biti "LPT" + string( lpt ) + ": " + if( ServerName = "" ,"","\\" + ServerName + "\" + QueueName )   1:
Detail ↑
Summary ↑
Footer ↑
```

d_password

The DataWindow d_password is used on w_set_password.

char(256)	UserName
char(48)	ServerName
char(48)	OldPassword
char(48)	NewPassword
char(48)	RetypePassword



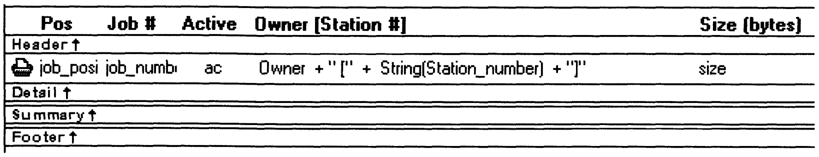
To change the password for the object:
username
On server:
servername

Please enter the following:
Old password: oldpassword
New password: newpassword
Retype new password: retypepassword

d_queue

The DataWindow d_queue is used on w_queue with the function f_check_queue.

number	Job_Position
number	Job_Number
char(1)	Active
char(48)	Owner
number	Station_Number
number	Size

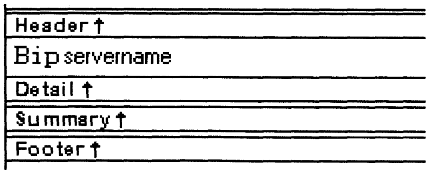


```
   Pos      Job #    Active   Owner [Station #]                                   Size (bytes)
Header ↑
   job_posi job_numbı   ac      Owner + " [" + String(Station_number) + "]"         size
Detail ↑
Summary ↑
Footer ↑
```

d_server_name_attached

The DataWindow d_server_name_attached is used on w_attachments with the function f_get_connections.

char(48)	ServerName
number	ServerID
number	DirectoryServices
number	Authenticated
char(1)	PrimaryConnection

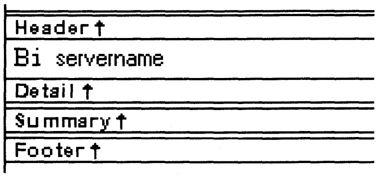


```
Header ↑
Bip servername
Detail ↑
Summary ↑
Footer ↑
```

d_server_name_unattached

The DataWindow d_server_name_unattached is used on w_attachments.

char(48)	ServerName
number	ServerID
number	DirectoryServices

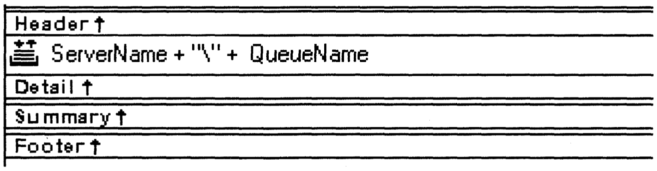


```
Header ↑
Bi servername
Detail ↑
Summary ↑
Footer ↑
```

d_server_queue_name

The DataWindow d_server_queue_name is used on w_printers with the function f_get_server_queues.

char(48)	ServerName
number	ServerID
char(48)	QueueName
number	QueueID

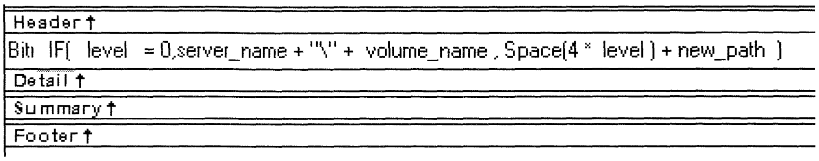


```
Header ↑
ServerName + "\" + QueueName
Detail ↑
Summary ↑
Footer ↑
```

d_server_volume

The DataWindow d_server_volume is used on w_map and the function f_get_server_volumes.

char(48)	Server_Name
number	Server_ID
char(18)	Volume_Name
number	Volume_ID
char(256)	Path
char(256)	New_Path
number	Dir_Handle
number	Level

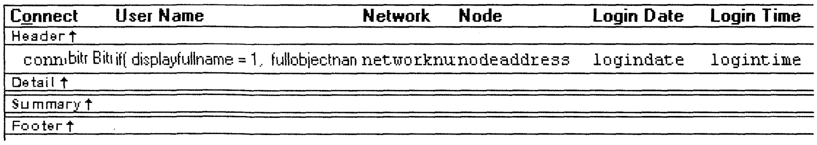


```
Header †
Biti IF( level = 0,server_name + "\" + volume_name , Space(4 * level ) + new_path )
Detail †
Summary †
Footer †
```

d_user_list

The DataWindow d_user_list is used on w_user_list with the function f_user_list.

number	ConnectID
char(48)	ObjectName
char(48)	FullObjectName
number	ObjectType
char(8)	NetworkNumber
char(12)	NodeAddress
date	LoginDate
time	LoginTime
number	DisplayFullName
number	CurrentConnection

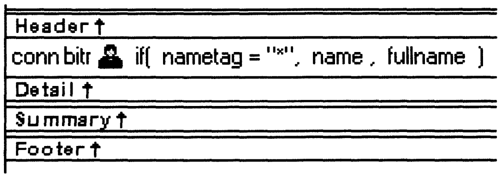


```
Connect    User Name                        Network   Node          Login Date   Login Time
Header †
  connıbitr Biti if( displayfullname = 1,  fullobjectnan networknu nodeaddress  logindate  logintime
Detail †
Summary †
Footer †
```

d_users

The DataWindow d_users is used on w_messages.

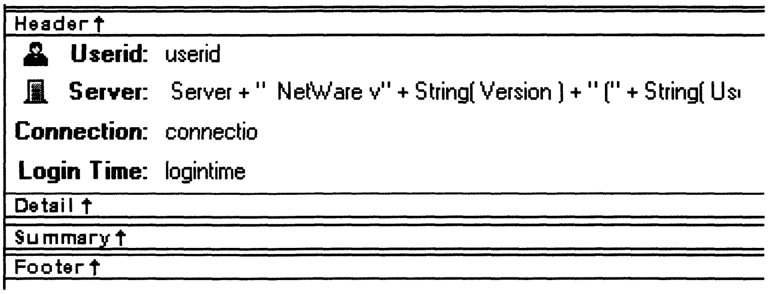
number	Connection
char(10)	Arrow
char(10)	Icon
char(1)	NameTag
char(25)	Name
char(30)	FullName


Header †
conn bitr if(nametag = "*", name , fullname)
Detail †
Summary †
Footer †

d_who_am_i

The DataWindow d_who_am_i is used on w_who_am_i with the function f_who_am_i.

char(48)	UserID
char(48)	Server
char(10)	Version
number	UserCount
number	Connection
datetime	LoginTime


Header †
Userid: userid
Server: Server + " NetWare v" + String(Version) + " (" + String(Us
Connection: connectio
Login Time: logintime
Detail †
Summary †
Footer †

CHAPTER 4

Functions

About this chapter

This chapter describes the functions used in the PowerBuilder Library for NetWare. Each description includes the syntax of the function, a description of the parameters, and an example.

Contents

The functions are listed alphabetically.

f_change_object_password

Syntax

f_change_object_password (*as_server_name, as_old_password, as_new_password*)

Parameter	Description
as_server_name	String identifying the server name. Pass by value
as_old_password	String identifying the old password. Pass by value
as_new_password	String identifying the new Password. Pass by value

Description

This function changes the object's password for the supplied bindery server. The function verifies that the old password is correct and then changes the password to the new password supplied. The user must already have a connection to the supplied bindery server.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code calls function f_change_object_password and passes the name of the bindery server the user wishes to change the password for, the old password, and the new password.

```
UINT  lui_result
STRING   ls_server_name
STRING   ls_old_password
STRING   ls_new_password

// Retrieve necessary information.
   ...
lui_result =
   f_change_object_password(ls_server_name,&
   ls_old_password,ls_new_password)

// If password change fails
IF lui_result <> 0 THEN
   ...
END IF
```

Required structures and external functions

s_connect_info
NWChangeObjectPassword
NWGetConnectionHandle
NWGetConnectionStatus
NWVerifyObjectPassword

f_char_hex_number_string

Syntax

f_char_hex_number_string (*as_char[ ]*)

Parameter	Description
as_char[]	Arrays of Chars passed by value

Description

This function converts an array of chars to a hex string.

Returns the hex conversion in the form of a string.

Return value

String.

Example

The following code returns a string containing the hexadecimal equivalent to the array of Chars passed into the function.

```
UINT  lui_result
UINT  lui_conn
UINT  lui_conn_id
CHAR  lc_internet_address[10]

// Get Internet Address.
lui_result = &
   nwGetInternetAddress(lui_conn,lui_conn_id,&
   lc_internet_address)

// Get internet address in hex string form.
ls_internet_address = &
   f_char_hex_number_string(lc_internet_address)
```

f_check_queue

Syntax

f_check_queue (*as_server_name,al_queue_id,as_queue*)

Parameter	Description	
as_server_name	String identifying the server name. Passed by value	
al_queue_id	Long identifying the queue ID. Passed by value	
as_queue	String containing the queue information listed below. Passed by reference	
	queue position	Position of print job in queue
	job number	Number of print job in queue
	active flag	Print job active flag
	client name	Owner or print job
	client station number	Location of print job owner
	job file size	Size of print job

Description

This function retrieves the contents of the specified printer queue for import into a DataWindow. The user must already have a connection to the supplied directory or bindery server.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code retrieves the contents of the specified server and queue and displays the information in a DataWindow based on d_queue.

```
STRING  ls_queue
STRING  ls_server_name
UINT  lui_queue_id
UINT  lui_resul

// Retrieve queue contents.
lui_result = &
   f_check_queue(ls_server_name,lul_queue_id,&
   ls_queue)
```

```
dw_queue.SetRedraw(FALSE)
dw_queue.Reset()

// If the queue is not empty, import the string.
IF Len(ls_queue) > 0 THEN
   dw_queue.ImportString(ls_queue)
END IF

dw_queue.SetRedraw(TRUE)
```

Required structures and external functions

s_NWQueueJobStruct
s_queueJobListReply
NWGetConnectionHandle
NWGetObjectName
NWGetQueueJobFileSize2
NWGetQueueJobList2
NWReadQueueJobEntry2

f_ds_change_object_password

Syntax

f_ds_change_object_password (*as_context, as_object_name, as_old_password, as_new_password*)

Parameter	Description
as_context	String identifying the directory context. Passed by value
as_object_name	String identifying the object name. Passed by value
as_old_password	String identifying the old password. Passed by value
as_new_password	String identifying the new password. Passed by value

Description

This function changes the authentication password for a directory object. The directory object must already be authenticated.

Returns 0 if successful.

Return value

Integer.

Example

The following code changes a directory service object's password for the supplied context.

```
UINT  lui_result
STRING   ls_context
STRING   ls_object_name
STRING   ls_old_password
STRING   ls_new_password

// Retrieve the necessary information to change the
// object's password.
   ...

lui_result = &
   f_ds_change_object_password(ls_context,&
   ls_object_name,ls_old_password,&
   ls_new_password)

// If password change fails
IF lui_result <> 0 THEN
   ...
END IF
```

Required external functions

NWDSChangeObjectPassword
NWDSCreateContext
NWDSFreeContext

NWDSSetContext
NWDSVerifyObjectPassword

f_ds_login

Syntax

f_ds_login (*as_object_name, as_name_context, as_password, as_tree*)

Parameter	Description
as_object_name	String identifying the object name. Passed by value
as_name_context	String identifying the name context. Passed by value
as_password	String identifying the object name's password. Passed by value
as_tree	String identifying the directory services tree. Passed by value

Description

This function logs in and authenticates an object in directory services using the supplied user name and password.

Returns 0 if successful.

Return value

Integer.

Example

The following code logs a user onto directory services with the supplied information.

```
UINT  lui_result
STRING   ls_user_name
STRING   ls_context
STRING   ls_password
STRING   ls_tree_name

// Retrieve the necessary parameters.
   ...

lui_result =
f_ds_login(ls_user_name,ls_context,ls_password,ls_tr
ee_name)

// If login failed
IF lui_result <> 0 THEN
   ...
END IF
```

Required external functions

NWDSAuditGetObjectID
NWDSAuthenticate
NWDSCreateContext
NWDSFreeContext
NWDSLogin
NWDSSetContext
NWGetNearestDirectoryService
NWGetPreferredServer
NWSetPreferredDSTree

f_ds_logout

Syntax

f_ds_logout (*as_context*)

Parameter	Description
as_context	String identifying the directory context. Pass by value

Description

This function logs an object out from directory services and all directory service servers. The user must already be authenticated on directory services.

Returns 0 if successful.

Return value

Integer.

Example

The following code logs the user out of directory services based on the specified context.

```
UINT  lui_result
STRING   ls_context

//Retrieve context to log out of.
   ...
lui_result = f_ds_logout(ls_context)

// If logout failed
IF lui_result <> 0 THEN
   ...
END IF
```

Required structures and external functions

s_connect_info
NWDSCreateContext
NWDSFreeContext
NWDSLogout
NWDSSetContext
NWFreeConnectionSlot
NWGetconnectionList
NWGetConnectionStatus
NWIsDSAuthenticated

f_ds_verify_password

Syntax

f_ds_verify_password (*as_context, as_object_name, as_password*)

Parameter	Description
as_context	String identifying the directory context. Pass by value
as_object_name	String identifying the object name. Pass by value
as_password	String identifying the password. Pass by value

Description

This function verifies the authentication password for a directory object. The user must already be authenticated on directory services.

Returns 0 if successful.

Return value

Integer.

Example

The following code verifies a directory services user password for the specified context.

```
UINT  lui_result
STRING   ls_context
STRING   ls_user_name
STRING   ls_password

// Retrieve necessary information.
   ...
lui_result =  &
   f_ds_verify_password(gs_context,ls_user_name, &
   ls_password)

// If verify password failed
IF lui_result <> 0 THEN
   ...
END IF
```

Required external functions

NWDSCreateContext
NWDSFreeContext
NWDSSetContext
NWDSVerifyObjectPassword

f_ds_who

Syntax

f_ds_who (*as_context, as_object_name*)

Parameter	Description
as_context	String identifying the directory context. Passed by value
as_object_name	String identifying the name of the logged in object. Passed by reference

Description

This function returns the distinguished name of the object currently logged in to directory services for the given context. The user must already be authenticated on directory services.

Returns 0 if successful.

Return value

Integer.

Example

The following code retrieves who the object name of the user on the default directory services server.

```
STRING   ls_server_name
STRING   ls_object_name
STRING   ls_tree
UINT  lui_conn
UINT  lui_result

// Check if the user is logged into NetWare.
IF f_logged_in_to_netware() <> 0 THEN
   ...
END IF

// Return the workstations default network
// connection handle.
lui_result = nwGetDefaultConnectionID(lui_conn)
IF lui_result <> 0 THEN
   ...
END IF

ls_tree = Space(256)

// Check presence or absence of Directory Services
// on the server.
gi_directory_services =
nwIsDSServer(lui_conn,ls_tree)

// If the server is a directory services server
IF gi_directory_services = 1 THEN
```

```
// Remove any trailing "_" from returned tree name
   gs_context = Left(ls_tree,Pos(ls_tree,"_") - 1)
   lui_result = f_ds_who(ls_tree,ls_object_name)

   // If could not return who
   IF lui_Result <> 0 THEN
      ...
   END IF
END IF
```

Required external functions

NWDSCreateContext
NWDSFreeContext
NWDSSetContext
NWDSWhoAmI

f_ds_who_am_i

Syntax

f_ds_who_am_i (*as_object_name, as_tree*)

Parameter	Description
as_object_name	String identifying the object's name. Passed by reference
as_tree	String identifying the current tree. Passed by reference

Description

This function retrieves the name and directory tree of the object currently logged into directory services. The user must already be authenticated on directory services.

Returns 0 if successful.

Return value

Integer.

Example

The following code checks whether the user is logged into the network. If yes, the code retrieves the user's object name and tree name.

```
INT   li_result
STRING   ls_who_am_i
STRING   ls_object_name
STRING   ls_tree
UINT  lui_num_conn

// Check if the user is logged into NetWare.
IF f_logged_in_to_netware() <> 0 THEN
   ...
END IF

// If user ID authenticated for directory
// services, retrieve DS who am i information.
li_result = nwIsDsAuthenticated()

IF li_result = 1 THEN
// Return the user's object name and current tree.
   li_result = f_ds_who_am_i(ls_object_name,ls_tree)

// If who am i failed
   IF li_result <> 0 THEN
      ...
   END IF
END IF
```

Required external functions

NWDSCreateContext
NWDSFreeContext
NWDSGetContext
NWDSWhoAmI
NWGetNearestDirectoryService
NWIsDSServer

f_get_connected_server_name

Syntax

f_get_connected_server_name (*as_server_names[]*, *aui_server_count*)

Parameter	Description
as_server_names[]	Array of strings identifying the connected server names. Pass by reference
aui_server_count	Unsigned Integer identifying the number of connected servers. Passed by reference

Description

This function retrieves an array of connected server names.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code retrieves a list of names for all server connections and adds them to a dropdown listbox if the user is logged on to the network.

```
UINT  lui_count
UINT  lui_server_count
UINT  lui_result
STRING   ls_server_names[]

// Check if the user is logged into NetWare.
IF f_logged_in_to_netware() <> 0 THEN
   ...
END IF

// Get server names and return them in an array.
lui_result = &
   f_get_connected_server_names(ls_server_names, &
   lui_server_count)

// If get connected server names failed
IF lui_result <> 0 THEN
   ...
END IF

// Loop through array to add server names to
// dropdown listbox.
For lui_count = 1 To lui_server_count
   ddlb_servers.AddItem(ls_server_names[lui_count])
Next
```

Required structures and external functions	s_connect_info NWGetConnectionList NWGetConnectionStatus NWGetMaximumConnections

f_get_connections

Syntax

f_get_connections (*as_connections*)

Parameter	Description
as_connections	String made up of the following tab-separated fields. Passed by reference for import into a data window

server name	String identifying the server name
connection handle	Unsigned Integer identifying the connection handle
authenticated flag	Integer identifying the authenticated flag (1 = True, 0 = False)
bindery services flag	Integer identifying the bindery services flag (1 = True, 0 = False)

Description

This function retrieves a list of information on connections to servers and bindery service objects.

Returns 0 if successful.

Return value

Integer.

Example

The following codes retrieves a string of connected servers and imports the string into a DataWindow based on d_server_name_attached.

```
STRING   ls_connections
UINT  lui_result

// Return a list of bindery and directory service
// connections.
lui_result = f_get_connections(ls_connections)

// If get connections failed
IF lui_result <> 0 THEN
   ...
END IF
dw_attached.SetRedraw(FALSE)
dw_attached.Reset()
dw_attached.ImportString(ls_connections)
dw_attached.SetRedraw(TRUE)
```

Required structures and external functions	s_connect_info
	NWGetconnectionList
	NWGetConnectionStatus
	NWGetMaximumConnections
	NWIsDSServer

f_get_context

Syntax

f_get_context (*as_context*)

Parameter	Description
as_context	String indicating directory context. Passed by reference

Description

This function retrieves the current (default) directory context.

Returns 0 if successful.

Return value

Integer.

Example

```
INT     li_result
STRING   ls_context

// Get the default directory context and display it
// in the window.
li_result = f_get_context(ls_context)

// If get context failed
IF li_result <> 0 THEN
   ...
END IF
```

Required external functions

NWDSCreateContext
NWDSFreeContext
NWDSGetContext

f_get_drives

Syntax

f_get_drives (*as_drive_map*)

Parameter	Description
as_drive_map	String made up of the following tab separated fields identifying the available drive mappings. Passed by reference drive number drive status connection handle root path relative path full path

Description

This function retrieves a string containing the available drive mappings for import into a DataWindow.

Returns 0 if successful.

Return value

Unsigned Integer.

Example

The following code retrieves a string of drive mappings and imports the string into a DataWindow based on d_drive_paths.

```
UINT  lui_result
STRING   ls_drive_map

// Return a string containing drive information.
lui_result = f_get_drives(ls_drive_map)

// If get drives failed
IF lui_result <> 0 THEN
   ...
END IF

// Fill the DataWindow with the string.
dw_data_drives.SetRedraw(FALSE)
dw_data_drives.Reset()
dw_data_drives.ImportString(ls_drive_map)
dw_data_drives.SetRedraw(TRUE)
```

Required external function

NWGetDriveStatus

f_get_login_time

Syntax

f_get_login_time (*aui_server_connection, aui_object_connection, adt_date_and_time*)

Parameter	Description
aui_server_connection	Unsigned Integer identifying the server connection handle. Passed by value
aui_object_connection	Unsigned Integer identifying the object's connection handle. Passed by value
adt_date_and_time	Datetime identifying the date and time. Passed by reference

Description

This function retrieves the login time for a given object's connection.

Returns 0 if successful.

Return value

Unsigned integer.

Required external function

NWGetConnectionInformation

f_get_lpt_list

Syntax

f_get_lpt_list (*as_lpt_connections*)

Parameter	Description	
as_lpt_connections	String made up of the following fields identifying the printer connections. Passed by reference	
	lpt device	lpt device number
	file server name	file server name
	connection	connection handle to file server
	queue name	print queue name
	queue id	print queue ID

Description

This function retrieves a string of printer connections for import into a DataWindow.

Returns 0 if successful.

Return value

Unsigned Integer.

Example

The following code retrieves a string containing the lpt connections and imports the string into a DataWindows based on d_lpt_server_queue_name.

```
UINT  lui_result
STRING   ls_lpt_connections

lui_result = f_get_lpt_list(ls_lpt_connections)

// If get lpt list failed
IF lui_result <> 0 THEN
   ...
END IF

dw_lpt1_server_name.SetRedraw(FALSE)
dw_lpt1_server_name.Reset()
dw_lpt1_server_name.ImportString(ls_lpt_connections)
dw_lpt1_server_name.SetRedraw(TRUE)
```

Required structures and external functions

s_connect_info
s_nwcapture_flags1
s_nwcapture_flags2
NWGetCaptureFlags

NWGetCaptureStatus
NWGetConnectionStatus
NWGetMaxPrinters
NWScanObject

f_get_server_queues

Syntax

f_get_server_queues (*s_server_queue*)

Parameter	Description	
s_server_queue	String made up of the following tab separated fields identifying the server and printer queue to retrieve info for. Passed by reference	
	server name	Server name
	connection handle	Server connection handle
	queue name	Print queue name
	queue id	Print queue ID

Description

This function retrieves the contents of a printer queue for the supplied server and queue.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code retrieves a string of server and queue names and imports the string into a DataWindow based on d_server_queue_name.

```
UINT  lui_return_value
STRING   ls_server_queue

lui_result = f_get_server_queues(ls_server_queue)

// If get server queues failed
IF lui_result <> 0 THEN
   ...

END IF
dw_server_queue_name.SetRedraw(FALSE)
dw_server_queue_name.Reset()
dw_server_queue_name.ImportString(ls_server_queue)
dw_server_queue_name.SetRedraw(TRUE)
```

Required structures and external functions

s_connect_info
NWGetconnectionList
NWGetConnectionStatus
NWGetMaximumConnections
NWScanObject

f_get_server_time

Syntax

f_get_server_time (*aui_server_connection, adt_date_and_time*)

Parameter	Description
aui_server_connection	Unsigned integer identifying the connection handle. Passed by value
adt_date_and_time	Datetime identifying the current date and time. Passed by reference

Description

This function retrieves the current date and time of the supplied server connection handle.

Returns 0 if successful.

Return value

Unsigned integer.

Required external function

NWGetFileServerDateAndTime

f_get_server_volumes

Syntax

f_get_server_volumes (*as_server_volumes*)

Parameter	Description
as_server_volumes	String made up of the following tab separated fields identifying volume names. Passed by reference

server name	Server name
connection handle	Server connection handle
volume name	Volume name
volume id	Volume ID
path	Path to display
new path	Parsed path
directory handle	Handle to directory
level	Level indicator for DataWindow

Description

This function retrieves a string containing volume names of the connected servers for import into a DataWindow.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code retrieves a string of server volumes and imports the string into a DataWindow based on d_server_volume.

```
UINT  lui_result
STRING   ls_server_volumes

// Return a string containing server volume
// information.
lui_result = f_get_server_volumes(ls_server_volumes)

// If get server volumes failed
IF lui <> 0 THEN
   ...
END IF

// Import the string into the DataWindow.
```

```
dw_server_volume.SetRedraw(FALSE)
dw_server_volume.Reset()
dw_server_volume.ImportString(ls_server_volumes)
dw_server_volume.SetRedraw(TRUE)
```

Required structures and external functions

s_connect_info
NWGetconnectionList
NWGetConnectionStatus
NWGetMaximumConnections
NWGetVolumeName
NWParseNetWarePath

f_is_client_running_netware

Syntax

f_is_client_running_netware (*as_version*)

Parameter	Description
as_version	String identifying the NetWare VLM version (such as 1.02). Passed by reference

Description

This function checks for NetWare running on the requester's system. This function ensures that the user is currently running the correct VLM version (1.01 or higher).

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code checks if Version 1.01 or higher of the NetWare VLM is running and halts the application if there is an error.

```
UINT  lui_result
Decimal  ld_version

SetPointer(HourGlass!)

// NetWare 4.0 requires > vlm 1.01.
ld_version = 1.01
lui_result = f_is_client_running_netware(ld_version)
IF lui_result <> 0 THEN
   HALT CLOSE
END IF
```

Required external function

NWGetRequesterVersion

f_logged_in_to_netware

Syntax	**f_logged_in_to_netware** ( )
Parameters	None.
Description	This function determines whether a workstation is logged into a network server. Returns 0 if successful.
Return value	Unsigned integer.
Example	The following code checks if the user is logged into a NetWare server. Useful to run before code that requires the user to be logged into NetWare.

```
// Check if the user is logged into NetWare.
IF f_logged_in_to_netware() <> 0 THEN
   ...
END IF
```

Required structures and external functions	s_connect_info NWGetConnectionStatus NWGetDefaultConnectionID

f_login

Syntax

f_login (*as_server_name,as_user_name,as_password*)

Parameter	Description
as_server_name	String identifying the server name. Passed by value
as_user_name	String identifying the user name. Passed by value
as_password	String identifying the password. Passed by value

Description

This function logs into a bindery service server with the supplied user name and password.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code logs the user into the specified bindery server.

```
UINT  lui_Result
STRING   lls_server_name
STRING   ls_user_name
STRING   ls_password

// Retrieve the necessary parameters.
lui_result = &
   f_login(ls_server_name,ls_user_name,ls_password)

// If login fails
IF lui_result <> 0 THEN
   ...
END IF
```

Required structures and external functions

s_connect_info
NWAttachToFileServer
NWGetConnectionHandle
NWGetConnectionStatus
NWLoginToFileServer

f_logout

Syntax

f_logout (*as_server_name*)

Parameter	Description
as_server_name	String identifying the server name. Passed by value

Description

This function logs out and detaches from a bindery services server.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code logs the user out of the specified bindery server.

```
UINT  lui_result
STRING   ls_server_name

// Retrieve server name.
   ...
lui_result = f_logout(ls_server_name)

// If logout failed
IF lui_result <> 0 THEN
   ...
END IF
```

Required external functions

NWDetachFromFileServer
NWGetConnectionHandle
NWLogoutFromFileServer

f_map

Syntax

f_map (*aui_drive, as_input_path*)

Parameter	Description
aui_drive	Unsigned integer identifying the drive number. Passed by value
as_input_path	String identifying the path to map. Passed by value

Description

This function maps the target drive to the specified directory path. The user must already have a connection to the specified directory path.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code maps a path to the specified drive number.

```
UINT  lui_drive
UINT  lui_result
STRING  ls_path

SetPointer(HourGlass!)

// Retrieve the drive to map to.
   ...

// Retrieve the drive path.
   ...

// Map the path to the drive.
lui_result = f_map(lui_drive, ls_path)

// If map failed
IF lui_result <> 0 THEN
   ...
END IF
```

Required external functions

NWDeleteDriveBase
NWGetDrivePath
NWGetDriveStatus
NWParseNetWarePath

f_netware_path_info

Syntax

f_netware_path_info (*as_path,as_server_name, as_object_name, ac_version, ai_user_count, ab_effective_rights[8]*)

Parameter	Description
as_path	String identifying the path to retrieve info for. Passed by reference
as_server_name	String identifying the server name for path. Passed by reference
as_object_name	String identifying the object name for the connection to the server. Passed by reference
ac_version	Decimal identifying the server version number. Passed by reference
ai_user_count	Integer identifying the maximum number of users allowed on the server at a time. Passed by reference
ab_effective_rights[8]	Boolean array of effective rights of the object on the server for the given path. Passed by reference

Description

This function retrieves NetWare information on the supplied network path. The user must already have a connection to the specified network path.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code displays a message box containing NetWare information on the effective rights of the user on the specified path.

```
UINT  lui_result
STRING   ls_path
STRING   ls_message
STRING   ls_message_temp
STRING   ls_server_name
STRING   ls_object_name
STRING   ls_effective_rights
DECIMAL  lc_version
UINT  lui_user_count
BOOLEAN lb_effective_rights[8]
STRING   ls_rights[8]
STRING   ls_rights_desc[8]
```

```
INT   li_count

// Fill arrays with descriptions.
ls_rights[] = {"A","F","M","E","C","","W","R"}
ls_rights_desc[] = {&
   "(A)~t~tChange Access Control.~n",&
   "(F)~t~tScan for Files.~n",&
   "(M)~t~tModify Directory.~n",&
   "(E)~t~tErase Directory.~n",&
   "(C)~t~tCreate Directories and Files.~n",&
   "",&
   "(W)~t*~tWrite to File.~n",&
   "(R)~t*~tRead from File.~n"}

// Get the selected path.
   ...

// Retrieve the NetWare path information.
lui_result = f_netware_path_info(ls_path,
ls_server_name, ls_object_name,&
   lc_version, lui_user_count, lb_effective_rights)

IF lui_result = 0 THEN
   ls_Message = "Server: " + ls_server_name + " is
running Netware " + &
      String(lc_version) + " (  " + &
      String(lui_user_count) + " user).~n" + &
      "Path: " + ls_path + "~nUser Name: " + &
      ls_object_name + &
      " ~n~nYour Effective Rights for this directory
are("

// Get the descriptions for the object's effective
// rights.
  FOR li_count = 8 TO 1 STEP -1
    IF lb_effective_rights[li_count] THEN
     ls_effective_rights = &
     ls_effective_rights + ls_rights[li_count]
     ls_message_temp = ls_message_temp +
ls_rights_desc[li_count]
    ELSE
     ls_effective_rights = ls_effective_rights + " "
      END IF
   NEXT
  // Set the message footer based on whether the
  // user has any effective rights.
  IF ls_effective_rights = "        " THEN
    ls_message =  ls_message + "        )~n" + &
      "Entries in Directory May Inherit (        )&
      rights.~nYou have " + &
      "NO RIGHTS to this Directory Area."
   ELSE
      ls_message =  ls_message + &
      ls_effective_rights + ")~n" + &
      ls_message_temp + "~n* Has no effect on "&
      + "directory~n~nEntries in "+&
```

```
            "Directory May Inherit &
            (" + ls_effective_rights + ") rights."
        END IF

        MessageBox("Drive Info...",ls_message,None!)
    END IF
```

Required structures and external functions

s_connect_info
NWGetConnectionHandle
NWGetConnectionStatus
NWGetEffectiveRights
NWGetFileServerInformation
NWParseNetWarePath

f_netware_server_info

Syntax

f_netware_server_info (*as_server_name, as_server_version, ac_object_name, aui_conn_id, ab_authenticated, ab_directory_services, ab_default_conn, ab_primary_conn, ab_preferred_conn*)

Parameter	Description
as_server_name	String identifying the server name to retrieve information for. Passed by value
ac_server_version	Decimal identifying the server version number. Passed by reference
as_object_name	String identifying the object connected to the server. Passed by reference
aui_conn_id	Unsigned Integer identifying the server connection number. Passed by reference
ab_authenticated	Boolean identifying the authenticated flag. Passed by reference
ab_directory_services	Boolean identifying the directory services flag. Passed by reference
ab_default_conn	Boolean identifying the current default connection flag. Passed by reference
ab_primary_conn	Boolean identifying the primary connection flag. Passed by reference
ab_preferred_conn	Boolean identifying the preferred directory services flag. Passed by reference

Description

This function retrieves NetWare server information for a user supplied server name. The user must already have a connection to the supplied server name.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code displays NetWare server information in a message box for the selected server.

```
BOOLEAN lb_authenticated
BOOLEAN lb_directory_services
BOOLEAN lb_default_conn
BOOLEAN lb_primary_conn
BOOLEAN lb_preferred_conn
DECIMAL  lc_version
STRING   ls_server_name
STRING   ls_object_name
STRING   ls_message
UINT  lui_result
UINT  lui_conn
STRING   ls_context
INT   li_directory_services
INT   li_authenticated

// Get selected server name.
   ...

// Retrieve NetWare server information.
lui_result = f_netware_server_info(ls_server_name, &
   lc_version,ls_object_name, lui_conn, &
   lb_authenticated, lb_directory_services,&
   lb_default_conn, &
   lb_primary_conn,lb_preferred_conn)

IF lui_result = 0 THEN
   // Set up message header.
   ls_message =
   "Server Name:~t" + ls_server_name + &
      "~nServerVersion:~t" + &
      String(lc_version) + "~nLogged in as:~t" + &
      ls_object_name + "~nConnection ID:~t" + &
      String(lui_conn) + "~nConnection Type:"

// Set server types: indicate if directory or
// bindery services.
   IF lb_directory_services THEN
      ls_message = ls_message + &
      "~n~tDirectory Services "
   ELSE
      ls_message = ls_message + "~n~tBindery "
   END IF

// Indicate if authenticated.
   IF lb_authenticated THEN
      ls_message = ls_message + "Authenticated"
   ELSE
      ls_message = ls_message + "Non-Authenticated"
   END IF

// Indicate if the default (current) connection.
   IF lb_default_conn THEN
      ls_message = ls_message + "~n~tCurrent Server"
   END IF

// Indicate if the primary connection.
   IF lb_primary_conn THEN
      ls_message = ls_message + "~n~tPrimary Server"
   END IF
```

```
// Indicate if the preferred DS connection.
   IF lb_preferred_conn THEN
      ls_message = ls_message + &
      "~n~tPreferred Server"
   END IF

   MessageBox("NetWare Info...",ls_message,None!)
END IF
```

Required structures and external functions

s_connect_info
NWGetConnectionHandle
NWGetConnectionStatus
NWGetDefaultConnectionID
NWGetFileServerInformation
NWGetPreferredDSServer
NWGetPrimaryConnectionID

f_user_list

Syntax

f_user_list (*as_server_name, as_user_list*)

Parameter	Description
as_server_name	String identifying the server name to retrieve information from. Passed by value
as_user_list	String containing the following user list information. Passed by reference

connection number	User connection number
object name	User object name
object full name	User object full name
object type	User object type
network number	Network number
node address	Node address
login date	User login date
login time	User login time

Description

This function retrieves a list of users currently logged into a given server. The user must already have a connection to the supplied server.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code retrieves a string containing a list of users for the specified server and imports the string into a DataWindow based on d_user_list.

```
UINT  lui_result
STRING   ls_user_list
// Retrieve server name.
   ...
// Retrieve the list of users on the server.
lui_result = &
   f_user_list(ls_server_name,ls_user_list)
IF lui_result = 0 THEN
   dw_user_list.SetRedraw(False)
```

```
      dw_user_list.Reset()
      dw_user_list.ImportString(ls_user_list)
      dw_user_list.SetRedraw(True)
   END IF
```

Required external functions

NWGetConnectionHandle
NWGetConnectionInformation
NWGetFileServerInformation
NWGetInternetAddress
NWReadPropertyValue

f_who_am_i

Syntax

f_who_am_i (*as_who_am_i*, *aui_num_conn*)

Parameter	Description
as_who_am_i	String containing the following workstation connection information. Passed by reference

object name	Object name
server name	Server name
version	Server version
maximum connections	Max number if connections on server
connection number	Connection number
login datetime	User login datetime

Parameter	Description
aui_num_conn	Unsigned integer identifying the number of connections returned. Passed by value

Description

This function returns a string containing connection information for the NetWare servers that the workstation is currently connected to.

Returns 0 if successful.

Return value

Unsigned integer.

Example

The following code retrieve a string containing the connection information for the user and imports the string into a DataWindow based on d_who_am_i.

```
UINT  lui_result
STRING   ls_who_am_i
UINT  lui_num_conn

li_result = f_who_am_i(ls_who_am_i,lui_num_conn)

dw_who_am_i.Reset()
dw_who_am_i.ImportString(ls_who_am_i)
```

Required structures and external functions

s_connect_info
NWGetConnectionInformation
NWGetconnectionList
NWGetConnectionStatus

NWGetFileServerInformation
NWGetMaximumConnections

CHAPTER 5

NetWare API External Function Calls

About this chapter	This chapter describes the NetWare external function calls used in the PowerBuilder Library for NetWare.
Contents	The functions are grouped by type and presented alphabetically.

Bindery services functions

Bindery service functions enable an application to maintain and manipulate the bindery of a file server. The bindery has been superseded by the NetWare Directory for NetWare 4.0.

NWChangeObjectPassword

Changes the specified object's password to a new password.

```
FUNCTION UINT NWChangeObjectPassword (
    UINT    Conn,
    STRING          ObjectName,
    UINT    ObjectType,
    STRING          OldPassword,
    STRING          NewPassword )
LIBRARY "NWCalls.DLL"
```

NWGetObjectName

Returns the name and object type of a bindery object on the network server.

```
FUNCTION UINT NWGetObjectName (
    UINT            Conn,
    ULONG           ObjectID,
    REF STRING      ObjectName,
    REF UINT                ObjectType )
LIBRARY "NWCalls.DLL"
```

NWIsObjectInSet

Searches a property of type SET for a specified object.

```
FUNCTION UINT NWIsObjectInSet (
    UINT    Conn,
    STRING          ObjectName,
    UINT    ObjectType,
    STRING          PropertyName,
    STRING          MemberName,
    UINT    MemberType )
LIBRARY "NWCalls.DLL"
```

NWReadPropertyValue

Reads the property value of a bindery object.

```
FUNCTION UINT NWReadPropertyValue (
     UINT            Conn,
     STRING          ObjectName,
     UINT            ObjectType,
     STRING          PropertyName,
     INT             DataSetIndex,
     REF STRING      DataBuffer,
     REF INT         MoreFlag,
                     REF INT    PropertyFlag )
LIBRARY "NWCalls.DLL"
```

NWScanObject

Searches for a bindery object name.

```
FUNCTION UINT NWScanObject (
     UINT            Conn,
     STRING          SearchName,
     UINT            SearchType,
     REF ULONG       ObjectID,
     REF STRING      ObjectName,
     REF UINT        ObjectType,
     REF INT         HasProperties,
     REF INT         ObjectFlags,
                     REF INT    ObjectSecurity )
LIBRARY "NWCalls.DLL"
```

NWVerifyObjectPassword

Verify the password of a bindery object on the specified NetWare server.

```
FUNCTION UINT NWVerifyObjectPassword (
     UINT            Conn,
     STRING          ObjectName,
     UINT            ObjectType,
     STRING          Password )
LIBRARY "NWCalls.DLL"
```

Connection services functions

Connection service functions are used to obtain information about the entities that are using the file server at a given time.

NWFreeConnectionSlot

Either removes all task dependencies on a task disconnect or completely tears down the connection for the system disconnect.

```
FUNCTION UINT NWFreeConnectionSlot (
     UINT    Conn,
     INT     DisconnectType )
LIBRARY "NWNet.DLL"
```

NWGetConnectionHandle

Returns the workstation's connection handle for the specified NetWare server.

```
FUNCTION UINT NWGetConnectionHandle (
     STRING      ServerName,
     UINT        Reserved1,
     REF UINT    Conn,
     STRING      Reserved2 )
LIBRARY "NWCalls.DLL"
```

NWGetConnectionInformation

Returns information about a logged-in object.

```
FUNCTION UINT NWGetConnectionInformation (
     UINT          Conn,
     UINT          ConnNumber,
     REF STRING    ObjectName,
     REF UINT      ObjectType,
     REF ULONG     ObjectID,
     REF CHAR      LoginTime[7] )
LIBRARY "NWCalls.DLL"
```

NWGetConnectionList

Returns a list of all connection handles.

```
FUNCTION UINT NWGetConnectionList (
     UINT           Mode,
     REF UINT       ConnListBuffer[],
     UINT           ConnListSize,
     REF UINT       numConnections )
LIBRARY "NWCalls.DLL"
```

NWGetConnectionNumber

Returns the connection number the requesting workstation uses to communicate with the NetWare server corresponding to the connection handle.

```
FUNCTION UINT NWGetConnectionNumber (
     UINT             Conn,
     REF UINT         ConnNumber )
LIBRARY "NWCalls.DLL"
```

NWGetConnectionStatus

Returns status information about a specified connection handle.

```
FUNCTION UINT NWGetConnectionStatus (
     UINT                  Conn,
     REF s_connect_info    ConnInfo,
     UINT                  ConnInfoSize )
LIBRARY "NWCalls.DLL"
```

NWGetDefaultConnectionID

Returns the default connection handle of the current session.

```
FUNCTION UINT NWGetDefaultConnectionID (
     REF UINT          Conn )
LIBRARY "NWCalls.DLL"
```

NWGetDefaultNameContext

Allows the caller to get the default name context.

```
FUNCTION UINT NWGetDefaultNameContext (
    UINT            BufferSize,
    REF STRING   Context )
LIBRARY "NWNet.DLL"
```

NWGetInternetAddress

Returns the internet address of the specified connection connNumber on the specified NetWare server.

```
FUNCTION UINT NWGetInternetAddress (
    UINT                Conn,
    UNIT                ConnNumber,
    REF CHAR         InternetAddress[10 ]
LIBRARY  "NWCalls.DLL"
```

NWGetMaximumConnections

Returns the maximum number of connections available at the requesting worsktation.

```
SUBROUTINE NWGetMaximumConnections (
    REF UINT          MaxConnections )
LIBRARY "NWCalls.DLL"
```

NWGetNearestDirectoryService

Returns a connection to the nearest Directory Services NetWare server (distance is determined by clock ticks).

```
FUNCTION UINT NWGetNearestDirectoryService (
    REF UINT    Conn )
LIBRARY "NWNet.DLL"
```

NWGetNextConnectionID

Gets the next connection in the VLM based on the connection handle passed.

```
FUNCTION UINT NWGetNextConnectionID (
    REF UINT    Conn )
LIBRARY "NWNet.DLL"
```

NWGetNumConnections

Returns the number of connections that can be supported by the VLM.

```
FUNCTION UINT NWGetNumConnections (
    REF UINT    NumConnections )
LIBRARY "NWNet.DLL"
```

NWGetObjectConnectionNumbers

Returns a list of connection numbers for clients logged in with the specified object name and type.

```
FUNCTION UINT NWGetObjectConnectionNumbers (
    UINT          Conn,
    STRING        ObjectName,
    UINT          ObjectType,
    REF UINT      NumConnections,
    REF UINT      ConnList,
    UINT          MaxConnections )
LIBRARY "NWCalls.DLL"
```

NWGetPreferredConnName

Returns the name of the preferred connection.

```
FUNCTION UINT NWGetPreferredConnName (
    REF STRING  PreferredName,
    REF UINT    PreferredType )
LIBRARY "NWNet.DLL"
```

NWGetPreferredDSServer

Returns the connection handle of the preferred directory server.

```
FUNCTION UINT NWGetPreferredDSServer (
    REF UINT    Conn )
LIBRARY "NWNet.DLL"
```

NWGetPreferredServer

Returns the preferred server.

```
FUNCTION UINT NWGetPreferredServer (
    REF UINT    Conn )
LIBRARY "NWCalls.DLL"
```

NWGetPrimaryConnectionID

Returns the workstation's primary network connection handle.

```
FUNCTION UINT NWGetPrimaryConnectionID (
    REF UINT    Conn )
LIBRARY "NWCalls.DLL"
```

NWIsDSAuthenticated

Returns whether Directory Services has credentials for a background authentication in the current DS tree.

```
FUNCTION UINT NWIsDSAuthenticated ( )
LIBRARY "NWNet.DLL"
```

NWIsDSServer

Checks presence or absence of Directory Services on the server.

```
FUNCTION INT NWIsDSServer (
    UINT            Conn,
    REF STRING      TreeName )
LIBRARY "NWNet.DLL"
```

NWIsIDInUse

Returns TRUE if the specified connection handle is in use.

```
FUNCTION UINT NWIsIDInUse (
    UINT     Conn )
LIBRARY "NWCalls.DLL"
```

NWSetPreferredDSTree

Sets the preferred Directory Server tree name in the requester's tables.

```
FUNCTION UINT NWSetPreferredDSTree (
    UINT         Length,
    STRING       Tree )
LIBRARY "NWNet.DLL"
```

Directory services functions

Directory services functions are used to access the NetWare directory and its related services. The directory supersedes the bindery.

NWDSAuditGetObjectID

Return a connection handle and an object ID for the object name relative to the context.

```
FUNCTION INT NWDSAuditGetObjectID (
    ULONG        Context,
    STRING       ObjectName,
    REF UINT     Conn,
    REF ULONG    ObjectID )
LIBRARY "NWNet.DLL"
```

NWDSAuthenticate

Establishes an authentication connection to a secured NetWare server using the unauthenticated connection and local data cached by calling NWDSLogin.

```
FUNCTION INT NWDSAuthenticate (
    UINT         Conn,
    ULONG        OptionsFlag,
    REF CHAR     SessionKey[16] )
LIBRARY "NWNet.DLL"
```

NWDSChangeObjectPassword

Changes the authentication password for a directory object.

```
FUNCTION INT NWDSChangeObjectPassword (
    ULONG      Context,
    ULONG      OptionFlags,
    STRING     ObjectName,
    STRING     OldPassword,
    STRING     NewPassword )
LIBRARY "NWNet.DLL"
```

NWDSCreateContext

Creates a directory context for directory client operations and initialize it to the default configuration.

```
FUNCTION ULONG NWDSCreateContext ( )
LIBRARY "NWNet.DLL"
```

NWDSFreeContext

Frees a previously allocated directory context variable.

```
FUNCTION INT NWDSFreeContext (
     ULONG        Context )
LIBRARY "NWNet.DLL"
```

NWDSGetContext

Returns a directory context variable.

```
FUNCTION INT NWDSGetContext (
     ULONG             Context,
     INT               Key,
     REF STRING        Value )
LIBRARY "NWNet.DLL"
```

NWDSLogin

Performs all authentication operations needed to establish a client's connection to the network and to the network's authentication service.

```
FUNCTION INT NWDSLogin (
     ULONG        Context,
     ULONG        OptionsFlag,
     STRING       ObjectName,
     STRING       Password,
     ULONG        ValidityPeriond )
LIBRARY "NWNet.DLL"
```

NWDSLogout

Terminate a client's connection to the network and invalidate any information cached locally by NWDSLogin.

```
FUNCTION INT NWDSLogout (
    ULONG Context )
LIBRARY "NWNet.DLL"
```

NWDSSetContext

Sets a directory context parameter of the directory context variable.

```
FUNCTION INT NWDSSetContext (
    ULONG      Context,
    INT        Key,
    STRING     Value )
LIBRARY "NWNet.DLL"
```

NWDSVerifyObjectPassword

Verifies the password of an object.

```
FUNCTION INT NWDSVerifyObjectPassword (
    ULONG      ContextHandle,
    ULONG      OptionsFlag,
    STRING     ObjectName,
    STRING     Password )
LIBRARY "NWNet.DLL"
```

NWDSWhoAmI

Returns the distiguished name of the object currently logged in.

```
FUNCTION INT NWDSWhoAmI (
    ULONG          Context,
    REF STRING     ObjectName )
LIBRARY "NWNet.DLL"
```

File server environment services functions

File server environment services functions enable applications to set certain file server parameters and return information about the servers.

NWAttachToFileServer

Attaches to the specified NetWare server.

```
FUNCTION UINT NWAttachToFileServer (
     STRING        ServerName,
     UINT          ScopeFlag,
     REF UINT      NewConn )
LIBRARY "NWCalls.DLL"
```

NWDetachFromFileServer

Breaks a workstation-NetWare server connection and relinquish the connection number.

```
FUNCTION UINT NWDetachFromFileServer (
     UINT     Conn )
LIBRARY "NWCalls.DLL"
```

NWGetFileServerDateAndTime

Returns the network date and time maintained on the specified NetWare server.

```
FUNCTION UINT NWGetFileServerDateAndTime (
     UINT              Conn,
     REF STRING        DateTimeBuffer )
LIBRARY "NWCalls.DLL"
```

NWGetFileServerInformation

Returns several items, including NetWare server name, NetWare versions, maximum and peak connections, number of connections currently in use, maximum volumes supported, SFT and TTS level of support.

```
FUNCTION UINT NWGetFileServerInformation (
    UINT    Conn,
    REF STRING  ServerName,
    REF CHAR    MajorVersion,
    REF CHAR    MinorVersion,
    REF CHAR    Revision,
    REF UINT    MaxConnections,
    REF UINT    MaxConnectionsUsed,
    REF UINT    ConnectionsInUse,
    REF UINT    NumVolumes,
    REF CHAR    SFTLevel,
    REF CHAR    TTSLevel )
LIBRARY "NWCalls.DLL"
```

NWGetFileServerName

Returns the name of the NetWare server based on conn.

```
FUNCTION UINT NWGetFileServerName (
    UINT        Conn,
    REF STRING  ServerName )
LIBRARY "NWCalls.DLL"
```

NWLoginToFileServer

Attempts to log an object on to the specified NetWare server.

```
FUNCTION UINT NWLoginToFileServer (
    UINT      Conn,
    STRING    ObjectName,
    UINT      ObjectType,
    STRING    Password )
LIBRARY "NWCalls.DLL"
```

NWLogoutFromFileServer

Attempts to log the workstation out of the specified NetWare server.

```
FUNCTION UINT NWLogoutFromFileServer (
     UINT     Conn )
LIBRARY "NWCalls.DLL"
```

File system services functions

File system services functions enable applications to manipulate extended file attributes, set and scan file information, and return information about volumes, directories, files, and more.

NWGetDirectoryHandlePath

Returns the path name of the directory associated with the given directory handle.

```
FUNCTION UINT NWGetDirectoryHandlePath (
     UINT            Conn,
     REF CHAR        DirHandle,
     STRING          DirPath )
LIBRARY "NWCalls.DLL"
```

NWGetEffectiveRights

Returns the caller's effective rights in the specified directory.

```
FUNCTION UINT NWGetEffectiveRights (
     UINT          Conn,
     INT           DirHandle,
     STRING        Path,
     REF UINT      EffectiveRights )
LIBRARY "NWCalls.DLL"
```

NWGetVolumeName

Returns the name of the volume associated with the specified volume number and the NetWare server conn.

```
FUNCTION INT NWGetVolumeName (
     UINT            Conn,
     UINT            VolNum,
     REF STRING      VolName )
LIBRARY "NWCalls.DLL"
```

NWScanDirectoryInformation2

Returns directory information for a directory specified by the connection handle, directory handle, and directory path.

```
FUNCTION UINT NWScanDirectoryInformation2 (
    UINT          Conn,
    INT           DirHandle,
    REF STRING    SearchDirPath,
    REF INT       SequenceNumber,
    REF STRING    DirName,
    REF ULONG     DirDateTime,
    REF ULONG     OwnerID,
    REF CHAR      MaximumRightsMax )
LIBRARY "NWCalls.DLL"
```

NWSetDirectoryHandlePath

Sets the target directory handle for the specified directory handle and path.

```
FUNCTION UINT NWSetDirectoryHandlePath (
    UINT          Conn,
    CHAR          SourceDirHandle,
    STRING        DirPath,
    REF CHAR      DestDirHandle )
LIBRARY "NWCalls.DLL"
```

Message services functions

Message services functions enable applications to send broadcast messages to specified target connections.

NWDisableBroadcasts

Informs the server that the client does not want to receive messages from other clients.

```
FUNCTION UINT NWDisableBroadcasts (
     UINT     Conn )
LIBRARY "NWCalls.DLL"
```

NWEnableBroadcasts

Allows a client to enable message reception after broadcast reception has been disabled using NWDisableBroadcasts.

```
FUNCTION UINT NWEnableBroadcasts (
     UINT     Conn )
LIBRARY "NWCalls.DLL"
```

NWGetBroadcastMessage

Returns a message from the server defined by conn.

```
FUNCTION UINT NWGetBroadcastMessage (
     UINT           Conn,
     REF STRING     Message )
LIBRARY "NWCalls.DLL"
```

NWGetBroadcastMode

Returns the receive message mode for the current workstation.

```
FUNCTION UINT NWGetBroadcastMode (
    UINT         Conn,
    REF UINT     Mode )
LIBRARY "NWCalls.DLL"
```

NWSendBroadcastMessage

Allows a client to send a broadcast message to the specified logical connections on the specified NetWare server.

```
FUNCTION UINT NWSendBroadcastMessage (
    UINT          Conn,
    STRING        Message,
    UINT          ConnCount,
    REF UINT      ConnList,
    REF STRING    ResultList )
LIBRARY "NWCalls.DLL"
```

NWSetBroadcastMode

Sets the message mode of the requesting workstation.

```
FUNCTION UINT NWSetBroadcastMode (
    UINT    Conn,
    UINT    Mode )
LIBRARY "NWCalls.DLL"
```

Miscellaneous services function

The one miscellaneous services function is NWGetRequesterVersion.

NWGetRequesterVersion

Returns the major version, minor version, and revision number of the OS requester or Shell.

```
FUNCTION UINT NWGetRequesterVersion (
    REF CHAR        MajorVersion,
    REF CHAR        MinorVersion,
    REF CHAR        Revision )
LIBRARY "NWCalls.DLL"
```

Path and drive services functions

Path and drive services functions map network drives, return drive information, and perform parsing on path strings.

NWDeleteDriveBase

Deletes a network drive mapping.

```
FUNCTION UINT NWDeleteDriveBase (
    UINT     DriveNumber,
    UINT     LocalScope )
LIBRARY "NWCalls.DLL"
```

NWGetDrivePath

Returns the drive path for the specified drive number.

```
FUNCTION UINT nwGetDrivePath (
    UINT             Drive,
    UINT             Mode,
    REF UINT         Conn,
    REF STRING       Path,
    REF UINT         Scope )
LIBRARY "NWCalls.DLL"
```

NWGetDriveStatus

Returns the status of drive number and, optionally, the associated connection and its path in various forms.

```
FUNCTION UINT NWGetDriveStatus (
    UINT             DriveNumber,
    UINT             PathFormat,
    REF UINT         Status,
    REF UINT         Conn,
    REF STRING       RootPath,
    REF STRING       RelativePath,
    REF STRING       FullPath )
LIBRARY "NWCalls.DLL"
```

NWParseNetWarePath

Parses a path and returns the connection handle, directory handle, and the new path to be used by subsequent NetWare requests.

```
FUNCTION UINT NWParseNetWarePath (
     REF STRING     Path,
     REF UINT       Conn,
     REF INT        DirHandle,
     REF STRING     NewPath )
LIBRARY "NWCalls.DLL"
```

NWSetDriveBase

Maps the target drive to the specified directory path.

```
FUNCTION UINT NWSetDriveBase (
     UINT          DriveNumber,
     UINT          Conn,
     INT           DirHandle,
     STRING        DirPath,
     UINT          DriveScope )
LIBRARY "NWCalls.DLL"
```

NWStripServerOffPath

Parses a server or volume path, copies the server name to the buffer specified by server, and returns the volume path.

```
FUNCTION STRING NWStripServerOffPath (
     STRING          Path,
     REF STRING      Server )
LIBRARY "NWCalls.DLL"
```

Print services functions

Print services functions let workstations redirect print jobs from local LPT devices to destinations on network NetWare servers.

NWEndCapture

Ends the capture for the specified LPT device.

```
FUNCTION UINT NWEndCapture (
     INT        LPTDevice )
LIBRARY "NWCalls.DLL"
```

NWFlushCapture

Allow the workstation software to flush a capture to the server, allowing a file to be printed.

```
FUNCTION UINT NWFlushCapture (
     INT        LPTDevice )
LIBRARY "NWCalls.DLL"
```

NWGetBannerUserName

Returns the user name printed on the banner pages for print jobs sent to any LPT device.

```
FUNCTION UINT NWGetBannerUserName (
     REF STRING      UserName )
LIBRARY "NWCalls.DLL"
```

NWGetCaptureFlags

Allows the application to get information concerning the capture.

```
FUNCTION UINT NWGetCaptureFlags (
    INT                        LPTDevice,
    REF s_NWcapture_flags1     CaptureFlagsRQ,
    REF s_NWcapture_flags2     CaptureFlagsRO )
LIBRARY "NWCalls.dll"
```

NWGetCaptureStatus

Returns whether a specific device is currently captures.

```
FUNCTION UINT NWGetCaptureStatus (
    INT      LPTDevice )
LIBRARY "NWCalls.DLL"
```

NWGetMaxPrinters

Returns the number of LPT ports for which captures can be managed.

```
FUNCTION UINT NWGetMaxPrinters (
    REF INT          NumPrinters )
LIBRARY "NWCalls.DLL"
```

NWSetBannerUserName

Sets the user name that is printed on banner pages for print jobs sent to LPT devices.

```
FUNCTION UINT NWSetBannerUserName (
    STRING           UserName )
LIBRARY "NWCalls.DLL"
```

NWSetCaptureFlags

Allows the application to set flags pertaining to the device redirection and print job information.

```
FUNCTION UINT NWSetCaptureFlags (
     INT                          Conn,
     INT                          LPTDevice,
     s_NWcapture_flags1           CaptureFlagsRW )
LIBRARY "NWCalls.DLL"
```

NWStartQueueCapture

Redirects a specified device to a queue.

```
FUNCTION UINT NWStartQueueCapture (
     UINT          Conn,
     INT           LPTDevice,
     ULONG         QueueID,
     STRING        QueueName )
LIBRARY "NWCalls.DLL"
```

Print server printer definitions services function

A print server printer definitions services function performs operations on printer form definitions in a PRINTDEF database.

There is just one such function.

NWPSPdfScanForm

Finds a form in PRINTDEF.

```
FUNCTION UINT NWPSPdfScanForm (
     UINT            ConnType,
     ULONG           ConnID,
     REF ULONG       Sequence,
     REF STRING      FormName )
LIBRARY "NWPsrv.DLL"
```

Queue management services functions

Queue management services functions allow applications to use queues to control the flow of jobs and services on the network.

NWChangeQueueJobPosition2

Changes a job's position in the queue.

```
FUNCTION UINT NWChangeQueueJobPosition2 (
     UINT          Conn,
     ULONG         QueueID,
     ULONG         JobNumber,
     ULONG         NewJobNumber )
LIBRARY "NWCalls.DLL"
```

NWGetQueueJobFileSize2

Returns the file size of the file associated with a queue entry.

```
FUNCTION UINT NWGetQueueJobFileSize2 (
     UINT               Conn,
     ULONG              QueueID,
     ULONG              JobNumber,
     REF ULONG          FileSize)
LIBRARY "NWCalls.DLL"
```

NWGetQueueJobList2

Returns a list of all jobs currently in the queue.

```
FUNCTION UINT NWGetQueueJobList2 (
     UINT                        Conn,
     ULONG                       QueueID,
     ULONG                       QueueStartPosition,
     REF s_queueJobListReply     job )
LIBRARY "NWCalls.DLL"
```

NWReadQueueJobEntry2

Allows an application to retrieve information about a job from a queue.

```
FUNCTION UINT NWReadQueueJobEntry2 (
     UINT                         Conn,
     ULONG                        QueueID,
     ULONG                        JobNumber,
     REF s_NWQueueJobStruct       job )
LIBRARY "NWCalls.DLL"
```

NWRemoveJobFromQueue2

Allows the workstation to remove a job from the queue.

```
FUNCTION UINT NWRemoveJobFromQueue2 (
     UINT        Conn,
     ULONG       QueueID,
     ULONG       JobNumber )
LIBRARY "NWCalls.DLL"
```

Index