Chapter 3
Bindery Design Specification

Abstract

Bindery.NLM provides Bindery name resolution and connection authentication for the 32-bit Client.

Contents

Introduction

Client32 supports several name services, including Bindery and NetWare Directory Services (NDS). Each of these name services is supported by a name service provider which interacts with the Name Service Multiplexor (NSMux) and ConnMan.

BINDERY.NLM is the name service provider that resolves bindery names and authenticates Bindery connections.

The Bindery module for Client32 will:

- Resolve NetWare Bindery names to a transport address or objectID
- Authenticate NetWare connections using bindery authentication
- Allow users to configure a preferred server

Like all Client32 Requester NLMs, Bindery is written entirely in C and can be dynamically loaded and unloaded.

Design Description

The Bindery module will register its services with both the Authentication Multiplexor (AuthMux) and the Name Services Multiplexor (NSMux), as shown in Figure 1. AuthMux multiplexes bindery authentication requests to the Bindery module, while the NSMux multiplexes name resolution requests to the Bindery module.

To register its authentication services with AUTHMux, Bindery uses **AUTHRegisterSvc.** Bindery calls **AUTHUnregisterSvc** to unregister its authentication services when BINDERY.NLM is unloaded.

Bindery will internally store (in encrypted form) authentication information for each connection so that bindery connections can be auto-reconnected.

Bindery provides name resolution services in order to resolve a NetWare Bindery server name to a transport address (or username to objectID) so that a connnection can be established.

To register its name resolution services with the NSMux, Bindery calls **NSMRegisterSvc**. When Bindery.NLM is being unloaded, it will call **NSMUnregisterSvc** to unregister its name service provider routines.

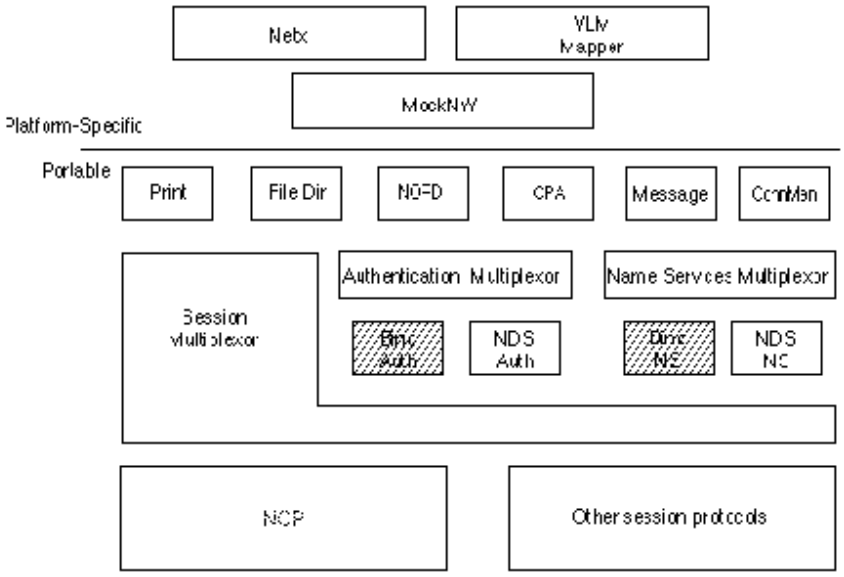

Figure 1. Bindery.NLM has two functional areas: authentication services (Bind Auth in figure) and name services (Bind NS in figure).

Bindery APIs

Bindery Authentication APIs

The following authentication service routines will be registered with AuthMux:

BinderyAuthenticateWithHandle	Authenticates a connection using a previously created handle.
BinderyCloseAuthenticationHandle	Closes the specified authentication handle.
BinderyCreateAuthenticationHandle	Creates an authentication handle that can be used to automatically authenticate connections.
BinderyGetAuthenticationInfo	Returns authentication information associated with a handle.
BinderyUnauthenticate	Unauthenticates a connection.

Bindery Name Service APIs

The following name service provider routines will be registered with the name service multiplexor for resolving NetWare Bindery names:

BinderyGetInitialConnection	Creates an initial Bindery connection.
BinderyGetPreferredServer	Gets a preferred server name for a particular scope.
BinderyQualifyConnectionMatch	Gets a preferred server name for a particular scope.
BinderyResolveIDToObject	Resolves a NetWare ID to a NetWare object name.
BinderyResolveNameToAddress	Resolves a server name to a transport address.
BinderyResolveObjectToId	Resolves an object name to a transport address/objectID pair.
BinderySetPreferredServer	Sets a preferred server name for a particular scope.

Constant Definitions

None specific to Bindery. See *conn.h* and *name_svc.h* constant definitions.

Structure Definitions

None specific to Bindery.

Return Codes

None specific to Bindery. See *conn.h* and *name_svc.h* return code definitions.

NESL Events

BINDERY.NLM will not produce any NESL events but will consume the following event produced by another Client32 module:

EVENT_CONN_DESTROYED

Meaning: ConnMan produces this event after calling a session protocol's **SessDisconnect** routine.

Action: BINDERY.NLM frees any resources associated with the destroyed connection, such as authentication information for auto-reconnection.

Configuration

Bindery will allow a user to configure the *preferred server* (i.e., the server to which it will initially attach when connecting to the network). The following keyword and server name will be read from NET.CFG by BINDERY.NLM when it loads:

PREFERRED SERVER = <server_name>

If a preferred server is specified, BINDERY.NLM will attempt to make a connection to the preferred server.

Deliverables

The following are the deliverables for Bindery:

Documents

BINDERYT.WP	Unit test plan that describes tests and expected results that the Bindery module must pass before being accepted for cross-platform integration testing.

Executables

BINDERY.NLM	NLM executable.

Source Code

BINDERY.C	C code that implements this design specification.
BINDERY.H	Header file that defines constants/structures defined in this design document.
BINDERY.MAK	Makefile to build BINDERY.NLM.
BINDERY.DEF	Definition file that defines link/build information for BINDERY.NLM.
BINDERY.IMP	File that lists external functions required by BINDERY.NLM.
BINDERY.EXP	File that lists functions that BINDERY.NLM will export for other modules to use.

Unit Test Executables

BINDERYT.NLM	The NLM application that tests BINDERY.NLM.

Unit Test Source

BINDERYT.C	C code that unit tests the BINDERY.NLM module.
BINDERYT.MAK	Makefile to build BINDERYT.NLM module.
BINDERYT.DEF	Definition file that defines link/build information for BINDERYT.NLM.
BINDERYT.IMP	File that lists external functions required by BINDERYT.NLM.