VMS Software

VSI OpenVMS

VSI Enterprise Directory Problem Solving Guide

Document Number: DO-DVEDPS-01A

Publication Date: May 2024

Operating System and Version: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-1H1 or higher

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

VSI Enterprise Directory Problem Solving Guide

Legal Notice

Preface

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

2. Intended Audience

This guide is intended for all users of the Enterprise Directory software and contains information that enables you to diagnose and solve problems with it.

3. Structure of This Guide

Chapter 1 describes the tools and utilities that you use to diagnose and solve Enterprise Directory problems.

Chapters 2 to 7 contain information about how to solve some common problems in a particular area of the Enterprise Directory, as follows:

- Chapter 2 explains how to solve problems with installing and configuring the Enterprise Directory, and with starting and stopping Enterprise Directory components.
- Chapter 3 explains how to solve problems with communications between Enterprise Directory components.
- Chapter 4 explains how to solve problems with distributed operations such as chaining and replication.
- Chapter 5 explains how to solve problems with manipulating directory entries and retrieving information from the directory.
- Chapter 6 explains how to solve problems with access control and authentication.
- Chapter 7 explains how to solve problems caused by insufficient system resources.

If you encounter a problem that is not described in these chapters, use the information in the rest of the book to diagnose and solve the problem.

Chapter 8 describes the accounting facility provided with this version of the Enterprise Directory.

Chapter 9 contains an explanation of each error and status message returned by NCL when managing Enterprise Directory entities, and by DXIM, and describes how to respond to these messages.

Chapter 10 describes the events and counters generated by the Enterprise Directory.

Appendix A contains a list of all the files installed and used by the VSI Enterprise Directory software, their locations and protections.

Appendix B contains a summary of the NCL directives that apply to Enterprise Directory entities.

4. Prerequisite Information

This guide assumes that you are familiar with the VSI Enterprise Directory software, and have read *VSI Enterprise Directory Management* and the release notes.

5. Related Documentation

You may find it useful when using this book to solve Enterprise Directory problems to have access to the information contained in the following:

- *VSI Enterprise Directory Management*
- *VSI OpenVMS Enterprise Directory Installing*

You might also find the following online help information useful:

- DXIM online help
- NCL Directory module online help

6. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at https://docs.vmssoftware.com.

7. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: <docinfo@vmssoftware.com>. Users who have VSI OpenVMS support contracts through VSI can contact <support@vmssoftware.com> for help with this product.

8. Typographical Conventions

this font	Introduces a new term or phrase.
`this typeface`	Indicates prompts and messages from the computer, and commands that you enter.
`$`	Indicates the prompt displayed by an OpenVMS system.

9. Abbreviations and Acronyms

ACIitem	Access Control Information Item
ASN.1	Abstract Syntax Notation 1
CCITT	See ITU-T
CDS	Cell Directory Service
DAP	Directory Access Protocol

DIB	Directory Information Base
DISP	Directory Information Shadowing Protocol
DIT	Directory Information Tree
DMD	Directory Management Domain
DOP	Directory Operational Binding Protocol
DSA	Directory System Agent
DSP	Directory System Protocol
DUA	Directory User Agent
DXIM	X.500 Information Management utility
EMA	Enterprise Management Architecture
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union-Telecom Standardization LDAP Lightweight Directory Access Protocol
MTA	Message Transfer Agent
NCL	Network Control Language
OSAK	OSI Applications Kernel
PDU	Protocol Data Unit
RDN	Relative Distinguished Name
ROSE	Remote Operations Service Element
XDS	X/Open Directory Services application programming interface

Chapter 1. Introduction to Problem Solving

This chapter describes the tools and utilities available to help you solve problems in your Enterprise Directory.

Problems can occur anywhere within the Enterprise Directory. They can occur during normal operation of the product, during specific operations such as reading or writing to the directory, or they can be the result of hardware failures or problems in the underlying software.

In most cases, you will be able to solve problems using the information in this guide. In others, you need to contact VSI.

If you have trouble solving a problem, refer to the release notes for up-to-date information on known problems.

VSI provides the following problem solving aids to help you:

- The Network Control Language
- Error and status messages
- Events
- DSA counters
- Diagnostic Trace utility
- Network isolation tool

In addition to these aids, you will also need all the worksheets and planning information you created while planning your directory.

1.1. Network Control Language

The Network Control Language (NCL) provides directives that you can use to monitor and control the configuration of a HP DSA.

The NCL online help includes a Directory_module topic which provides a comprehensive description of all NCL directives that relate to a HP DSA.

Using these NCL directives, you can:

- Manipulate entities and subentities

  Appendix B gives a summary of the NCL directives that you can use to manipulate entities and subentities of Enterprise Directory.

- Manipulate status and characteristic attributes

  Each entity and subentity has attributes that allow you to monitor and control the operation of the entity or subentity. These attributes give information that is useful during problem solving and

provide you with a means through which you can configure the Enterprise Directory to suit your particular needs.

Appendix B gives a summary of status and characteristic attributes for the DSA entity and its subentities.

- Manipulate knowledge information

  Each knowledge reference maintained by a DSA is implemented as a subentity of that DSA, thereby ensuring that you can monitor and control knowledge references using NCL directives. This allows you to delete old knowledge references, create new ones, or change existing ones. For example, you can delete an old subordinate reference and create a new one simply by deleting the Subordinate Reference subentity and creating a new one.

- Display DSA counters

  The DSA entity maintains a number of counters that can be displayed using the NCL SHOW directive. Refer to Section 1.4 for more information.

1.1.1. Remote Management Using TCP/IP

The NCL director can be used to manage entities on remote systems. This remote management requires the use of DECnet-Plus. If you want to manage a DSA remotely, but your local system does not run DECnet-Plus, you may need to use the following command to make the NCL utility use the TCP/IP protocol instead:

```
ncl> SET NCL TRANSPORT TCPIP
```

1.2. Error and Status Messages

Error and status messages are the first pointer to problems within the Enterprise Directory. Most error messages relate to problems in the command specified by a user, for example, invalid names for entries or invalid attribute types. Additionally however, errors can be returned from violations of security policy, schema rules and service controls, and to indicate problems with directory functions.

Chapter 9 describes all error and status messages returned by Enterprise Directory. Error messages are returned by the DXIM management utility and the NCL director.

1.3. Events

Events are a useful means of diagnosing Enterprise Directory problems.

Chapter 10 describes all events generated by Enterprise Directory. Event dispatching is automatic on OpenVMS systems.

1.4. DSA Counters

DSA counters are incremented automatically by the DSA. All counters are automatically set to zero when you create a DSA entity. You cannot change their values. Use the NCL SHOW DSA directive to display the values of the counters.

For counters to be useful, you must monitor their values regularly over a period of time. This allows you to determine normal system behavior.

By monitoring counters regularly, you can determine peak periods of activity and problem areas within the Enterprise Directory. For example, a counter with a value that normally increases slowly but suddenly starts to increase rapidly might indicate a problem.

Chapter 10 describes counters in more detail.

1.5. OSAK Trace Utility

You can use the OSAK trace utility to trace information transferred on an association between a DUA and a DSA, or between two DSAs. Traced information shows the protocol exchange between the two applications and any data transferred between them.

Generally, protocol errors should not occur on associations between HP DUAs and DSAs, unless there is an error in the DUA or DSA software. However, they could occur when you are trying to connect to another vendor's DSA or DUA. For information about decoding and analyzing the protocol trace, see *OSI Application Developer's Toolkit OSAK Programming*.

Do not enable the recording of protocol information permanently, as this consumes disk space and could slow down the system on which tracing is being performed. Only enable the recording of protocol information when you are trying to find out why an association to or from a specific DUA or DSA has failed.

1.5.1. Starting a Trace

To enable OSAK tracing, edit the DSA startup file, DXD$DIRECTORY:DXD $DSA_STARTUP_INPUT.COM, and remove the comment marker (!) from the following two lines:

```
$ ! define/process osak_trace on
$ ! define/process osak_trace_file dxd $directory:osak_trace.bin
```

Then, rerun the startup file and create and enable the DSA entity.

This enables the OSAK trace. OSAK will now trace all incoming connections and outgoing connections on this node.

1.5.2. Stopping a Trace

To disable OSAK tracing, edit the DXD$DIRECTORY:DXD$DSA_STARTUP_INPUT.COM file, and comment out the two lines that apply to OSAK trace. Then rerun the startup file, and create and enable the DSA entity.

1.6. DSA Worksheets and Planning Information

The DSA worksheets and general planning information created during the planning phase of configuring your Enterprise Directory, are a valuable source of information for problem solving.

Such information provides details on, for example:

- Design guidelines and naming policies for your directory
- The identity of DSAs containing master copies of directory entries

- The identity of DSAs containing shadow copies of directory entries
- The identity of DSAs acting as supplier DSAs
- The identity of DSAs acting as consumer DSAs
- Any subordinate references created
- Any superior references created
- The identity and location of naming contexts
- The access control rights established for each naming context Refer to *VSI Enterprise Directory Management* for details.

1.7. Network Isolation Tool

The network isolation tool enables you to test a network connection between two applications. The network isolation tool attempts to establish a network connection to a node where you have installed an application, for example a DSA. The network isolation tool identifies any errors that prevent the connection being established and can also provide a detailed description of each stage of the network connection.

Refer to Section 3.6 for details of how to run the network isolation tool.

Chapter 2. Problems with Installation, Configuration and Startup

This chapter covers problems associated with:

- Installing the Enterprise Directory software (see Section 2.1)
- Running the DSA configuration procedure (see Section 2.2)
- Configuring DSA Attributes Manually (see Section 2.3)
- Starting the DSA or DSA Not Running (see Section 2.4)
- Stopping the DSA (see Section 2.5)
- Running the DUA configuration procedure (see Section 2.6)
- Starting or running DXIM (see Section 2.7 and Section 2.8)
- Lookup CLient Problems (see Section 2.9)

2.1. Installing the Enterprise Directory Software

Installation of the Enterprise Directory is performed using the VMSINSTAL utility.

The table below lists typical installation problems, and refers to the section where the solution to each problem is described.

Table 2.1. Problems with Installation

Symptom	Refer to
Directory Cannot Be Found	Section 2.1.1
The Subset Requires a Specific Operating System Version	Section 2.1.2
The Subset Requires Another Subset to Be Installed	Section 2.1.3
DECnet-Plus NCL Dictionary Could Not Be Updated	Section 2.1.4
DECnet-Plus Help Could Not Be Updated	Section 2.1.5
Failed to Rename MAILbus 400 MTA File	Section 2.1.6
No Valid Enterprise Directory Database Found or Error Occurred	Section 2.1.7
CML Configuration File Could Not Be Updated	Section 2.1.8
Incorrect Operating System Version	Section 2.1.9
DECnet Not Installed or Incorrect Version	Section 2.1.10
Incorrect OSAK Version	Section 2.1.11

Symptom	Refer to
DECwindows Motif Not Installed or Incorrect Version	Section 2.1.12
A DSA Is Already Running on This Node	Section 2.1.13
Insufficient Disk Space	Section 2.1.14
Insufficient Free Global Pages	Section 2.1.15
Insufficient Free Global Sections	Section 2.1.16
DXD$SERVER Account Not Found	Section 2.1.17
Incorrect Ordering of Installation Subsets	Section 2.1.18
License Not Installed	Section 2.1.19
Incorrect Privileges	Section 2.1.20
IVP Returns Errors or Warnings	Section 2.1.21

2.1.1. Directory Cannot Be Found

This error only occurs during upgrade installations or reinstallations. It indicates that Enterprise Directory logical names have been found, but that they do not refer to accessible directory specifications.

There can be two errors: one for the Enterprise Directory work directory, and another for the accounting directory.

The installation proceeds, and eventually prompts you for the device that the Enterprise Directory work directory should be created on. The installation creates the work directory on that disk. It also amends SYS$STARTUP:DXD$LOGICALS_STARTUP.COM to make sure that the logical name DXD $DIRECTORY is defined during Directory Service startup. If the accounting directory could not be found, then a new accounting directory is created as a subdirectory of the work directory.

If the installation cannot find the directories because the relevant disk(s) are temporarily unavailable, terminate the installation, and do not run the installation until the disk(s) are available.

2.1.2. The Subset Requires a Specific Operating System Version

`The` *`name`* `subset requires that` *`op_sys_version`* `be installed on this system`

where *name* is the name of a subset, and *op_sys_version* identifies the prerequisite version of the operating system.

The subset cannot be installed because the system is running the wrong version of the operating system. Install the correct prerequisites before attempting to install the Enterprise Directory.

2.1.3. The Subset Requires Another Subset to Be Installed

`The` *`name`* `subset requires that` *`subset`* `be installed on this system`

where ***name*** is an Enterprise Directory subset and ***subset*** is the prerequisite subset. Install that subset before attempting to install the Enterprise Directory.

2.1.4. DECnet-Plus NCL Dictionary Could Not Be Updated

```
Enterprise Directory Base :- DECnet-Plus NCL Dictionary could not be
updated - missing DECnet-Plus files
```

Reinstall DECnet-Plus before attempting to install the Enterprise Directory.

2.1.5. DECnet-Plus Help Could Not Be Updated

```
Enterprise Directory Base :- DECnet-Plus help could not be updated -
missing DECnet-Plus files
```

Reinstall DECnet-Plus before attempting to install the Enterprise Directory.

2.1.6. Failed to Rename MAILbus 400 MTA File

```
Enterprise Directory Server :- Failed to rename MAILbus 400 MTA snapshot file

Enterprise Directory Server :- Failed to rename MAILbus 400 MTA update file

Enterprise Directory Server :- Failed to rename MAILbus 400 MTA schema file

Enterprise Directory Server :- Failed to rename MAILbus 400 MTA snapid file
```

The installation has detected the presence of a MAILbus 400 MTA Version 1.0 database, and has failed to rename the file. These errors can only be returned if you have not performed the mandatory upgrade of Version 1.0 of the MAILbus 400 MTA, which was only available on ULTRIX and OpenVMS systems. Refer to the MAILbus 400 MTA documentation for details of the mandatory update tasks.

2.1.7. No Valid Enterprise Directory Database Found or Error Occurred

```
Enterprise Directory Server :- No valid database found or error
occurred - cannot continue
```

The installation has detected that existing database files are corrupt, and therefore the installation terminates. Delete or move the existing X.500 database files before installing the Enterprise Directory.

2.1.8. CML Configuration File Could Not Be Updated

```
Enterprise Directory Server :- file could not be updated - missing
DECnet file. This is a fatal error
```

where ***file*** is the name of the CML configuration file. The installation attempts to edit this file so it is possible to use NCL to manage the DSA. If this attempt fails, the installation terminates.

Reinstall DECnet-Plus before attempting to install the Enterprise Directory.

2.1.9. Incorrect Operating System Version

Upgrade your operating system version to that indicated by the installation procedure and rerun the installation.

2.1.10. DECnet Not Installed or Incorrect Version

DECnet software is not installed or you do not have the correct version of DECnet software needed to install the Enterprise Directory product. Upgrade your version of DECnet software to that indicated by the installation procedure and rerun the installation.

2.1.11. Incorrect OSAK Version

You do not have the correct version of OSAK software needed to install the Enterprise Directory product. OSAK software is a component of DECnet software. Upgrade your version of OSAK software to that indicated by the installation procedure and rerun the installation.

2.1.12. DECwindows Motif Not Installed or Incorrect Version

If you intend to use the DXIM Motif interface, you must install DECwindows Motif.

Install the version of DECwindows Motif indicated by the installation procedure.

2.1.13. A DSA Is Already Running on This Node

The Enterprise Directory cannot be installed on a node if a DSA is already running on that node. Stop the DSA and rerun the installation procedure.

Stop the DSA by entering the following command:

```
$ @SYS$STARTUP:DXD$COMMON_SHUTDOWN
```

2.1.14. Insufficient Disk Space

There is not enough free disk space available to continue with the installation. Make more space available and rerun the installation procedure or use an alternate working device.

Refer to the **setld** or VMSINSTAL documentation for more advice about installing software products.

2.1.15. Insufficient Free Global Pages

There are insufficient global pages (GBLPAGES) available to install the Enterprise Directory software.

Increase the number of global pages available by the amount indicated by the installation procedure, reboot your system and then rerun the installation. Refer to the OpenVMS system management documentation for information on how to modify the global pages system parameter.

2.1.16. Insufficient Free Global Sections

There are insufficient global sections (GBLSECTIONS) available to install the Enterprise Directory product.

Increase the number of global sections available by the amount indicated by the installation procedure, reboot your system and then rerun the installation. Refer to the OpenVMS system management documentation for information on how to modify the global sections system parameter.

2.1.17. DXD$SERVER Account Not Found

The DXD$SERVER account does not exist. This account is needed by the DSA server process.

The installation procedure creates a new DXD$SERVER account and the installation proceeds. No action is necessary.

2.1.18. Incorrect Ordering of Installation Subsets

The Enterprise Directory subsets must be installed in the correct order. The Enterprise Directory Base subset must be installed first. All other subsets are dependent on the Base subset.

When you delete the subsets, the reverse order applies. The Base subset must be deleted last. If you try to delete this subset while other subsets are still installed, you are asked to confirm your actions. If you continue to delete the Enterprise Directory Base subset, the remaining subsets may not operate correctly.

2.1.19. License Not Installed

Check that you have the correct license installed by typing the following command:

```
root> lmf list
```

Refer to the Software Product Description for more information about licenses.

2.1.20. Incorrect Privileges

To install the Enterprise Directory product, you require privileges.

Ensure that the VMSINSTAL utility from a suitably privileged account such as the SYSTEM account

2.1.21. IVP Returns Errors or Warnings

The installation verification procedure (IVP) checks that the files required for the Enterprise Directory components you have installed are present in the correct location and have the correct protection. If the IVP cannot find a file, or finds that the file protection is not correct, it displays a message indicating the problem. It also displays a message if it cannot find a component that has previously been installed.

You can run the IVP at any time, by typing the following command:

```
$ @SYS$TEST:DXD$IVP*.COM
```

Note that an IVP failure message indicates that an Enterprise Directory file has changed since installation. This does not necessarily mean that your Enterprise Directory will not work, but you are recommended to investigate any changes reported by the IVP.

2.2. Running the DSA Configuration Procedure

The DSA configuration procedure sets a DSA's Presentation Address attribute. This saves you from having to plan and set the attribute manually.

The procedure also sets the AE Title attribute so that you can start the DSA. However, the procedure only sets a temporary value which you should replace with a planned value, as described in *VSI Enterprise Directory Management*.

This section discusses problems running the DSA configuration procedure. Section 2.3 discusses problems with configuring DSA attributes manually.

Table 2.2. Problems Running the DSA Configuration Procedure

Symptom	Refer to
Configuration Procedure Not Found	Section 2.2.1
Insufficient Privileges to Run the Utility	Section 2.2.2
The DSA Cannot be Configured While It Is Running	Section 2.2.3
You Must be a Superuser to Run the Utility	Section 2.2.4
There is No DSA Installed on this Node to Configure	Section 2.2.5
Failed to Create the DSA. Cannot Configure the DSA	Section 2.2.6
Error Configuring Presentation Address. Rerun this Utility	Section 2.2.7
Error Configuring LDAP Port. Rerun this Utility	Section 2.2.8
Error Configuring AE Title. Rerun this Utility	Section 2.2.9
Failed to Delete the DSA. The DSA is Still in State OFF	Section 2.2.10
Cannot Construct a Presentation Address. Cannot configure DSA	Section 2.2.11
DECnet-Plus and RFC1006 are not configured	Section 2.2.12

2.2.1. Configuration Procedure Not Found

Check that the configuration procedure, DXD$DSA_CONFIGURE.COM, is in the correct location and has the correct file protection. See Appendix A for details of the correct location and protection.

2.2.2. Insufficient Privileges to Run the Utility

You have insufficient privileges to run the configuration procedure.

Run the configuration procedure from an account that has the SYSPRV, OPER and BYPASS privileges.

2.2.3. The DSA Cannot be Configured While It Is Running

If the DSA is in state ON, it cannot be configured. However, if it is in state ON, it must already have a valid configuration. If you really want to reconfigure the DSA's presentation address, disable the DSA before running the configuration procedure. The procedure will set the DSA's presentation address, but will not reset the DSA's AE Title. The procedure assumes that any existing AE Title is better than the temporary value it would set, and therefore does not change the existing value.

2.2.4. You Must be a Superuser to Run the Utility

You are not logged in as superuser and so cannot run the configuration procedure.

2.2.5. There is No DSA Installed on this Node to Configure

Check that the DSA component is installed.

2.2.6. Failed to Create the DSA. Cannot Configure the DSA

The DSA cannot be created or configured. Check DSA resources and configuration.

2.2.7. Error Configuring Presentation Address. Rerun this Utility

An error has occurred when configuring the Presentation Address.

Rerun the configuration procedure, or optionally run the appropriate NCL command for checking purposes.

2.2.8. Error Configuring LDAP Port. Rerun this Utility.

An error has occurred when configuring the LDAP port.

Rerun the configuration procedure, or optionally run the appropriate NCL command for checking purposes.

2.2.9. Error Configuring AE Title. Rerun this Utility.

An error has occurred when configuring the AE Title.

Rerun the configuration procedure, or optionally run the appropriate NCL command for checking purposes.

2.2.10. Failed to Delete the DSA. The DSA is Still in State OFF

The DSA cannot be deleted.

Run the appropriate NCL command for checking purposes.

2.2.11. Cannot Construct a Presentation Address. Cannot configure DSA

The DSA cannot be configured and a Presentation Address cannot be constructed.

Check the DSA selectors and NSAPs are correctly set.

2.2.12. DECnet-Plus and RFC1006 are not configured

Check that DECnet-Plus is configured for RFC1006.

2.3. Configuring DSA Attributes Manually

The table below shows the errors that can occur when manually configuring DSA attributes.

Table 2.3. Problems Configuring the DSA Manually

Symptom	Refer to
The DSA Is in the Wrong State	Section 2.3.1
Invalid Attribute Value in AE Title	Section 2.3.2
Invalid Attribute Value in Presentation Address	Section 2.3.3

2.3.1. The DSA Is in the Wrong State

This error only occurs if you try to specify a Presentation Address or AE Title for the DSA when the DSA is in state ON. Other attributes can be configured in state ON or state OFF.

The DSA can only be in state ON if it already has a valid Presentation Address and AE Title. If you really mean to reconfigure either of these attributes, use the DISABLE DSA command to set the DSA state to OFF. Then reconfigure the DSA Presentation Address or AE Title as required. Note that the Enterprise Directory provides a DSA configuration utility. Section 2.3.3 explains how to run the utility. When you have reconfigured the DSA, use the ENABLE DSA command to set the DSA state to ON so that the DSA can receive user requests.

If you have changed the DSA's Presentation Address or AE Title, and other DSAs have knowledge references that specify this DSA, then you need to reconfigure those other DSAs so that their knowledge references reflect the change to this DSA's configuration details. If other DSAs have out of date knowledge references, the operation of your Enterprise Directory will be impaired.

2.3.2. Invalid Attribute Value in AE Title

The application entity title (AE Title) is badly formed. An AE Title must conform to the following rules:

- It must be specified using distinguished name syntax, for example, `/c=US/o=Abacus/cn=DSA2`
- It must not contain an incorrect attribute name keyword, for example, `/county=utah`
- The attribute values must be of the correct syntax. For example, `/c=UK` is not valid because "UK" is not a valid country code.
- You must use printable string characters. Valid characters are:
  - '( )+:-./=?
  - The space character

- 0123456789
- abcdefghijklmnopqrstuvwxyz
- ABCDEFGHIJKLMNOPQRSTUVWXYZ

Use quotation marks to enclose a string that contains any of the following characters:

- A space character
- comma (,)
- equals sign (=)
- question mark (?)
- slash (/)
- quotation mark (", ‘, or ')

For example, the following distinguished name contains several RDNs, each of which must be quoted because of the characters within them:

```
/o="Smith, Jones, and Brown Inc."/ou="Sales/Mktg"/cn="Dan 'Swifty' DSA"
```

For more information on AE Titles, refer to *VSI Enterprise Directory Management* or the NCL online help.

2.3.3. Invalid Attribute Value in Presentation Address

You have specified an incorrect presentation address. Use the DSA configuration utility to set the presentation address.

You need SYSPRV and OPER privileges to run the configuration utility. To run the utility, type:

```
$ @SYS$STARTUP:DXD$DSA_CONFIGURE
```

If your privileges are insufficient, the utility displays an error message and exits.

2.4. Starting the DSA or DSA Not Running

Use the NCL CREATE DSA and ENABLE DSA directives to start a DSA. You can either enter these directives manually, or run the NCL script SYS$STARTUP:DXD$DSA_STARTUP.NCL. The DSA server process, DXD$DSA_SERVER, must be running before you can create or enable a DSA entity.

The DSA can also be started automatically as part of the system startup. The Enterprise Directory installation procedure adds the necessary commands to the system startup files. On an OpenVMS system, the commands are preceded by comment flags. These comment flags must be removed if you want the DSA to start automatically at system startup.

If the DSA is not running, or it fails to start after a system reboot, check that the comment flags have been removed from the system startup file. If they have, it may be that the system startup procedure has failed. In this case, try starting the DSA using NCL, by either directly entering the directives, or by using

the DSA startup script file as input. If you receive an error, refer to the table below for information about correcting the error.

Table 2.4. Problems Starting the DSA

Symptom	Refer to
Error Sending Command Request	Section 2.4.1
The DSA Information Tree is Corrupt	Section 2.4.4
The DSA Information Tree is Incompatible with this Version of the DSA	Section 2.4.3
The DSA Information Tree and Schema are Incompatible	Section 2.4.4
The Schema is Corrupt	Section 2.4.5
The Schema is Incompatible with this Version of the DSA	Section 2.4.6
No Resource Available	Section 2.4.7
The DSA's AE Title Attribute Has Not Been Set	Section 2.4.8
The DSA's Presentation Address Attribute Has Not Been Set	Section 2.4.9
The License Check Has Failed For This Product	Section 2.4.10
The DSA Entity Already Exists	Section 2.4.11
The DSA Cannot Open the Database	Section 2.4.12
The DSA is Currently Being Created	Section 2.4.13
The DSA Does Not Support the Specified Option	Section 2.4.14

2.4.1. Error Sending Command Request

This occurs because a CREATE DSA command is issued very quickly after a DELETE DSA command, or because the DSA server process has not been started.

Wait for a few seconds, and then repeat the CREATE DSA command. If this does not work, execute the DSA startup procedure as follows:

```
$ @SYS$STARTUP:DXD$DSA_STARTUP.COM
```

Then issue a CREATE DSA command.

2.4.2. The DSA Information Tree is Corrupt

The DSA did not start because it could not read its DIB fragment from disk into memory.

This can be caused by the following problems:

- The Enterprise Directory logical names are incorrectly defined, such that DSA cannot find the database files.

  If the logical names are missing, or point to the wrong directory, or to multiple directories, use SYS $STARTUP:DXD$COMMON_STARTUP.COM to define the logicals correctly.

- The file protection of the database files has been changed.

The protection is listed in Appendix A. If it has been changed, reset it to match the documented protection, and then create the DSA.

- The database files really are corrupt.

  If none of the above reasons explain the error, then the database files might really be corrupted. This is very unlikely, but if it occurs you can:

  i. Move the database files to a safe place

  ii. Restore a recent backup of the database files

  iii. Contact VSI if you want your corrupt database files investigated

For details of how to backup your database files, see Section 5.7.

If you do not have a usable backup, you must delete the DSA database, and start again.

2.4.3. The DSA Information Tree is Incompatible with this Version of the DSA

The DSA failed to read its DIB fragment from disk into memory. This is due to the DIB fragment being incompatible with the software version of the DSA.

This should not happen, because each version of the DSA provides backwards compatibility, and automatically makes any conversions required to its database.

If this error occurs, and you have a backup copy of the database files, restore the database files, and try to start the DSA again. If this still fails, report the problem to VSI.

2.4.4. The DSA Information Tree and Schema are Incompatible

This error returns one of four diagnostic statements:

- Attribute OID `oid` missing from the schema.
- Structure Rule `n` missing from the schema.
- Object class OID `oid` missing from the schema
- Attribute `attr` should have syntax `syn`

where `oid` is an object identifier, such as { 2 5 4 60}, and `n` is an integer.

In all cases, the message means that the DIT contains examples of information that is not defined in the schema.

The two most likely explanations for these problems are that:

- You have edited the schema and deleted the relevant definitions
- You have reinstalled the DSA, and have not remembered to recompile the schema to build in any customizations.

Every time you install the DSA, a default schema is installed. You need to make sure that any customizations are compiled back into the schema after every installation.

Old versions of the schema are saved in a subdirectory of DXD$DIRECTORY during every installation. If you have accide ntally deleted some definitions, or you have reinstalled the DSA, you should find that your customized files can be retrieved from that subdirectory.

2.4.5. The Schema is Corrupt

Check that the Enterprise Directory logicals are correctly defined, and have only one definition. If the Enterprise Directory logicals have multiple definitions, the DSA can report this error when it fails to find the DSA information tree files. To ensure that the Enterprise Directory logicals are correctly defined, run the following startup procedure:

```
$ @SYS$STARTUP:DXD$COMMON_STARTUP.COM
```

Check that the DXD$SCHEMA.DAT file is installed and in the directory defined by the logical name DXD$DIRECTORY and has the correct protection (see Appendix A for details of the correct file protection).

If none of the above solves the problem, recompile the schema. If the schema is missing, reinstall the Enterprise Directory.

2.4.6. The Schema is Incompatible with this Version of the DSA

The schema version differ s from the DSA version. Each version of the DSA includes new schema files and a new schema compiler. If the DSA tries to read a schema that was compiled by an old version of the schema compiler, it can report this error. Recompile the schema and restart the DSA, as follows:

```
$ SET DEFAULT DXD$DIRECTORY
$ RUN SYS$SYSTEM:DXD$SCHEMA_COMPILER.EXE
```

2.4.7. No Resource Available

Either the NCL CREATE DSA or the ENABLE DSA directive has failed due to a lack of system resources.

If you are attempting to enable the DSA for the first time, then this error can be caused if you have not created the transport templates required by the Enterprise Directory.

Refer to the installation documentation for details of this mandatory post-installation task.

If the DSA is installed on an OpenVMS system, or the templates have been created, refer to Chapter 7 for more information.

The unavailability of the DSA's LDAP port will generate this error. Check that another application is not using the TCP/IP port assigned to the DSA for LDAP. You can configure the DSA to use a different TCP/IP port for LDAP.

2.4.8. The DSA's AE Title Attribute Has Not Been Set

The DSA could not be enabled because it has not been assigned a valid application entity (AE) title.

Assign the DSA a valid AE title by using the NCL SET DSA directive. For more information on application entity titles, refer to *VSI Enterprise Directory Management* or the NCL online help.

2.4.9. The DSA's Presentation Address Attribute Has Not Been Set

The DSA could not be enabled because it has not been assigned a valid presentation address.

Use the DSA configuration utility to set a valid presentation address.

2.4.10. The License Check Has Failed For This Product

The DSA could not be created because a valid Enterprise Directory Server license has not been installed.

Check which licenses you have installed as follows:

```
root> lmf list
```

Install the appropriate license and restart the DSA. Refer to the Software Product Description for more information about licenses.

2.4.11. The DSA Entity Already Exists

A DSA could not be created because a DSA is already running on this system. Enable this DSA for communication using the NCL ENABLE directive.

2.4.12. The DSA Cannot Open the Database

A DSA could not open the database because the database is being used by another DSA on another node on the cluster. Only one DSA can access a

database at any one time. When the DSA loads the database, the database is locked to prevent another DSA accessing it.

To find the DSA in use, look in file DSA_informationtree.lock in the DXD$DIRECTORY directory.

2.4.13. The DSA is Currently Being Created

A DSA is already being created.

Wait for the DSA to finish creating.

2.4.14. The DSA Does Not Support the Specified Option

The DSA does not support the specified option. Upgrade to the latest version of the product.

2.5. Stopping the DSA

You can stop a DSA using the NCL DISABLE DSA and DELETE DSA directives, or by running the NCL script file DXD$DSA_S HUTDOWN.NCL.

If you cannot stop the DSA with NCL, you will have to stop the DSA process manually. Before doing this, check that the DSA is not merely taking a long time to write the DSA DIB fragment out to disk. This is done in response to the NCL DELETE DSA directive.

To stop the DSA process manually, enter the **SHOW SYSTEM** command to obtain the process_id of the DXD$DSA-S ERVER process, and then enter the comman d **STOP/ID=*process_id*** where ***process_id*** is the process_id obtained from the **SHOW SYSTEM** command.

2.6. Running the DUA Configuration Procedure

To configure the application defaults for DXIM, run the DUA configuration procedure, DXD $DUA_CONFIGURE.COM. This section describes the errors that can occur when you run the DUA configuration procedure.

The table below shows typical configuration problems.

Table 2.5. Problems with DUA Configuration Procedure

Symptom	Refer to
Configuration Procedure Not Found	Section 2.2.1
Insufficient Privileges to Run the Utility	Section 2.6.2
DXD$DIRECTORY Not Defined	Section 2.6.3
Unable to Obtain DUA Defaults from DSA	Section 2.6.4
Cannot Write DUA Defaults File	Section 2.6.5
The Node Name is Unreachable or Does Not Exist	Section 2.6.6
Unable to Bind to the DSA Over RFC1006	Section 2.6.7
You Must be a Superuser to Run the Utility	Section 2.6.8
Warning: RFC 1006 is Installed but the Kernal Needs Rebuilding	Section 2.6.9
Neither DECnet-Plus nor RFC1006 are Installed. Aborting.	Section 2.6.10

2.6.1. Configuration Procedure Not Found

Check that the configuration procedure, DXD$DUA_CONFIGURE.COM, is in the correct location and has the correct file protection. See Appendix A for details of the correct location and protection.

2.6.2. Insufficient Privileges to Run the Utility

You have insufficient privileges to run the configuration procedure.

Run the configuration procedure from an account that has the SYSPRV and OPER privileges.

2.6.3. DXD$DIRECTORY Not Defined

The logical name DXD$DIRECTORY is not defined.

To define the logical name, run the DXD common startup file as follows:

```
$ @SYS$STARTUP:DXD$COMMON_STARTUP.COM
```

Rerun the configuration procedure.

2.6.4. Unable to Obtain DUA Defaults from DSA

The DUA configuration procedure is unable to contact the DSA to obtain the default DUA configuration information. This is due to one of the following problems.

- NCL is unable to communicate with the DSA on the specified node.

  Check the DSA on the remote node is in state ON. To check the state of the DSA on the remote node, ***node***, issue the following NCL directive on the node where you are running the DUA configuration procedure:

  ```
  ncl> show node node dsa state
  ```

- The DSA has not been created or started.

  The DSA should be in state ON. If it is not, create or start the DSA as described in Section 2.4. To check the state of the DSA, issue the following NCL directive on the node where the DSA is installed.

  ```
  ncl> show dsa state
  ```

- The DSA has not been configured.

  If the DSA has been configured, it will have an AE title and presentation address defined. If the DSA has not been configured, configure it as described in *VSI Enterprise Directory Management*.

  To check whether the DSA has been configured, issue the following NCL directive on the node where the DSA is installed.

  ```
  ncl> show dsa ae title, presentation address
  ```

- The utility used RFC1006 to make the connection to the DSA (because DECnet-Plus was not available or the NCL SHOW DSA directive failed), and the connection has been rejected by the DSA for security reasons.

  If the utility uses RFC1006, then it is attempting to bind to the DSA, rather than use an NCL directive. In this case, it must satisfy the security requirements of the DSA, but has failed to do so.

  Check the values of the following characteristic attributes of the DSA: Writer Names, Reader Names, Writer NSAPs, Reader NSAPs. If any of these exist, amend the value of the Reader NSAPs attribute to include the RFC1006 NSAP of the system that you are running the configuration utility on. This enables the DSA to recognize the utility, and allow connections. You will need privileges to amend the characteristic attributes of a DSA. Refer to the NCL help for the Directory_Module for details of the Reader NSAP attribute.

2.6.5. Cannot Write DUA Defaults File

The DUA configuration procedure cannot write the DUA defaults file to disk. This may be because there is insufficient disk space, or because the directory protections are incorrectly set.

2.6.6. The Node Name is Unreachable or Does Not Exist

The DUA configuration procedure does not recognize the name of the node that you specified. Find out the correct name of the node where the DSA is installed. Rerun the DUA configuration procedure and specify the correct node name.

2.6.7. Unable to Bind to the DSA Over RFC1006

The DUA configuration procedure is unable to bind to the DSA using RFC1006. This could be due to one of the following problems:

- The DSA is not created or enabled

  Check the state of the DSA by issuing the following NCL directive on the node where the DSA is installed.

  ```
  ncl> show dsa state
  ```

  The DSA should be in state ON. If it is not, contact the person managing the DSA and ensure that the DSA is created and enabled. Rerun the DUA configuration procedure after the DSA is enabled.

- The DSA's Presentation Address might be invalid

  If the DSA is in the ON state, then check that the DSA's Presentation Address contains an RFC1006 NSAP. To display the DSA's Presentation Address, issue the following NCL directive on the node where the DSA is installed:

  ```
  ncl> show dsa presentation address
  ```

  Take the appropriate action as follows:

  - The DSA's Presentation Address does not contain an RFC1006 NSAP Find out if RFC1006 is supported on the node where the DSA is installed.

    If RFC1006 is supported on the node, then contact the person managing the DSA and request that the DSA's Presentation Address is set to include the RFC1006 network address. Rerun the DUA configuration procedure after the DSA has been reconfigured.

    If the node where the DSA is installed is unable to support RFC1006, then you need to have DECnet-Plus running on the node where you are running the DUA configuration procedure. Rerun the DUA configuration procedure when DECnet-Plus is available.

  - The DSA's Presentation Address contains an RFC1006 NSAP

    Validate the DSA's presentation address. To validate the DSA's presentation address, invoke DXIM on the node where you are running the DUA configuration procedure. Refer to *VSI Enterprise Directory Management* for information about how to invoke DXIM. Use the DXIM bind command to connect to the DSA. Specify the DSA's presentation address in the bind command, for example:

    ```
    dxim> bind to address paddr
    ```

    where ***paddr*** is the DSA's presentation address.

    If DXIM is able to bind to the DSA, then the DSA's presentation address contains a valid RFC1006 NSAP. You need to enter the DSA's presentation address in the DUA defaults file

on the node where you are running the DUA configuration procedure. Edit the DUA defaults file and change the **DUA.KnownDSAs.paddr** entry so that it specifies the correct presentation address for the DSA. Refer to *VSI Enterprise Directory Management* for information about how to determine a presentation address.

If DXIM is unable to bind to the DSA, then contact the person managing the DSA and confirm that the presentation address you are using is correct for the node where the DSA is installed.

- A network error

  If the DSA's presentation address is correct, then the problem is due either to a network failure or because the RFC1006 daemon is not running on the node where the DSA is installed.

  Refer to Chapter 3 for information about solving problems with network connections and how to start the RFC1006 daemon.

2.6.8. You Must be a Superuser to Run the Utility

You are not logged in as superuser and so cannot run the configuration procedure.

2.6.9. Warning: RFC 1006 is Installed but the Kernal Needs Rebuilding

Configure RFC1006 after the configuration procedure has completed.

2.6.10. Neither DECnet-Plus nor RFC1006 are Installed. Aborting.

Install DECnet-Plus and RFC1006.

2.7. Starting DXIM

Table 2.6 lists problems that apply to starting both the DXIM command line interface and the DXIM Motif interface. These or similar problems might also occur with other X.500 client applications. In such cases, the solutions suggested in the relevant sections may also help you with such client applications.

If any of the problems are caused by missing files or files that have not been installed, you must re-install the Enterprise Directory product as described in the installation documentation.

Table 2.6 shows typical problems encountered while starting DXIM.

Table 2.6. Problems Starting DXIM

Symptom	Refer to
DXIM Command Not Found or Not Recognized	Section 2.7.1
Error Activating Image	Section 2.7.2
DXIM Cannot Open the Schema File	Section 2.7.3
DXIM Cannot Read the Schema File	Section 2.7.4
DXIM Cannot Open the UID File	Section 2.7.5
DSA Is Unavailable (Motif Interface Only)	Section 2.7.6

Symptom	Refer to
Unable to Communicate with DSA	Section 2.7.7

2.7.1. DXIM Command Not Found or Not Recognized

The DXIM default startup script file cannot be foun d or the DXIM DCL verb is not recognized by the operating system. Check that:

- The command is being recognized.

  To ensure that OpenVMS recognizes the newly installed DXIM application, log out of the system and log back in.

- The appropriate files are installed and in the correct directory.

  Ensure that the DXIM executable images are installed and in the correct location.

- The file has the correct protection as specified in Appendix A.

2.7.2. Error Activating Image

If the file specified in the error message is DXD$DXIM_CLI.EXE or DXD$DXIM_MOTIF.EXE, an executable image for the DXIM command line interface or the DXIM Motif interface has not been found. Check that these images are installed in the correct directories and have the correct file protection set. See Appendix A for information about the file locations and protection.

If the file specified in the error message is DECW$XLIBSHR.EXE, check that the DECwindows Motif is installed correctly.

2.7.3. DXIM Cannot Open the Schema File

The schema binary file cannot be foun d or cannot be read.

Check that the logical name DXD$DIRECTORY is defined. If the logical name is not defined, run the DXD common startup file as follows:

```
$ @SYS$STARTUP:DXD$COMMON_STARTUP.COM
```

Check that the DXD$SCHEMA.DAT file is installed and in the directory defined by the logical name DXD$DIRECTORY and has the correct protection. See Appendix A for details of the correct file protection.

If the schema file is in the correct location and has the correct file protection, check that it is not corrupt. If necessary, recompile the schema source files and create a new binary file. Refer to *VSI Enterprise Directory Management* for more information on how to recompile the schema source files.

2.7.4. DXIM Cannot Read the Schema File

DXIM cannot read the schema binary file (DXD$SCHEMA.DAT) because it is in an invalid format. The software version of DXIM and the schema binary file do not match.

Recompile the schema source files and create a new binary file. Refer to *VSI Enterprise Directory Management* for more information on how to recompile the schema source files.

2.7.5. DXIM Cannot Open the UID File

The DXIM Motif user interface definition (UID) DXD$DXIM.UID could not be opened. Check that the UID file is installed in the correct directory and has the correct file protection. See Appendix A for details of the correct file location and protection.

2.7.6. DSA Is Unavailable (Motif Interface Only)

This is due to DXIM being unable to bind to the DSA. Refer to Section 3.1 for information on how to recover from bind problems.

2.7.7. Unable to Communicate with DSA

If you start DXIM, but receive an **Unable to Communicate with DSA** error, you are not correctly bound to the DSA. See Section 3.1.

2.8. DXIM Not Operating as Expected

This is due to one of the reasons shown in Table 2.7.

Table 2.7. Problems Running DXIM

Symptom	Refer to
Initial Entry Set to the Root Entry (Motif Interface Only)	Section 2.8.1
Incorrect Browse or Search Base (Motif Interface Only)	Section 2.8.2
DXIM Initialization File Has Not Been Run (Command Line Interface Only)	Section 2.8.3

2.8.1. Initial Entry Set to the Root Entry (Motif Interface Only)

DXIM uses two DUA defaults files at startup. One, located in the /usr/etc directory or the directory defined by the system logical name DXD$DIRECTORY, is for system-wide definitions. The other, located in your home or SYS$LOGIN directory, allows you to customize DXIM to suit your own particular needs. The values specified in the DUA defaults file located in your home or SYS$LOGIN directory override the values specified in the system-wide DUA defaults file.

On startup, DXIM requires an initial entry that it can use as the browse base for the Browse window and the search base for the Find window. DXIM looks for the following, using the first entry it finds as the initial entry:

`DUA.InitialEntry` in the users local defaults file
`DUA.InitialEntry` in the system-wide defaults file
`DUA.DomainRoot` in the users local defaults file
`DUA.DomainRoot` in the system-wide defaults file

Refer to *VSI Enterprise Directory Management* for more information on the DUA defaults file.

If the initial entry has invalid syntax, it is still displayed, but an error message is also displayed.

Ensure that either the initial entry parameter or domain root parameter is set to point to a valid directory entry. If neither of these parameters is set, DXIM defaults to the root of the DIT ("/").

2.8.2. Incorrect Browse or Search Base (Motif Interface Only)

If you start DXIM and the browse or search base is not what you expected, check to ensure that you have correctly specified the entry in the `DUA.InitialEntry` or the `DUA.DomainRoot` parameter of the DUA defaults file in your home or SYS$LOGIN directory. If you do not have a copy of the

DUA defaults file in your home or SYS$LOGIN directory, or you have not set the parameters in that file, check that the values specified in the system-wide DUA defaults file are correct.

2.8.3. DXIM Initialization File Has Not Been Run (Command Line Interface Only)

The DXIM command line interface can use an initialization file that contains DXIM comman ds that can be executed when the utility is invoked. The file is called .dximrc and must be in SYS $LOGIN:DXD$DXIM.INI. If DXIM cannot find this file, it simply assumes that there is no file to be foun d, and does not return an error message.

If your initialization file is not being run, check that it is in your home directory and has the correct file protection (see Appendix A for details of file protection).

2.9. Lookup Client Problems

Enterprise Directory provides a user application: the X.500 Lookup Client. This section discusses problems that might occur when you are configuring or using the Lookup Client.

Table 2.8 lists the problem areas.

Table 2.8. Problems Using the Lookup Client

Symptom	Refer to
Configuring the Lookup Client	Section 2.9.1
Starting the Lookup Client	Section 2.9.2
Using the Lookup Client	Section 2.9.3
Displaying the Lookup Client Motif Interface	Section 2.9.4
Using the Lookup Client Help	Section 2.9.5

2.9.1. Configuring the Lookup Client

If you have problems running the Lookup Client configuration procedure, check the following:

- The configuration procedure is in the correct location and has the correct file protection. The procedure is called SYS$STARTUP:DXD$LUC_CONFIGURE.COM. See Appendix A for details of the correct protections.
- You have sufficient privileges to run the procedure. Use an account with SYSPRV and OPER privileges.

- There is sufficient disk space to write the defaults file to its directory, and that the directory protection permits files to the written. The target directory is SYS$SYSTEM.

2.9.2. Starting the Lookup Client

If you have problems starting the Lookup Client, check the following:

- If you have defined a foreign command to invoke the Lookup Client, check the definition of the foreign command.
- Check that the Lookup Client files are in their correct locations and have the correct protections (see Appendix A). The relevant files are:

```
SYS$SYSTEM:DXD$LOOKUP_MOTIF.EXE
SYS$SYSTEM:DXD$LOOKUP_CLI.EXE
SYS$SYSTEM:DXDLU.DEFAULTS
SYS$MESSAGE:DXD$LUC_MSG.EXE
DECW$SYSTEM_DEFAULTS:DXDLU.UID
```

2.9.3. Using the Lookup Client

The behaviour of the Lookup Client is defined by the defaults file, which provides a description of the settings available.

2.9.4. Displaying the Lookup Client Motif Interface

If you are running the Lookup Client Motif interface on a remote system, you need to make sure that the display is directed to your local system. If Lookup Client displays an error indicating that it cannot open the display, use the SET DISPLAY command.

2.9.5. Using the Lookup Client Help

The Lookup Client Motif interface uses the Bookreader utility to display its help. If you do not have Bookreader on your system, the help cannot be accessed. Refer to your system manager to get Bookreader installed.

The Lookup Client command line interface on OpenVMS systems reads a help library called SYS $HELP:DXDLU.HLB. If you have problems using the help, check the location and protection of this library (see Appendix A for details of the correct protection).

Chapter 3. Problems with Communications

This chapter covers problems with connections between a DUA and a DSA, and between two DSAs. It includes, for example, binding problems, OSAK problems, transport problems and network problems.

Connections are established using any of the following protocols:

- The Directory Access Protocol (DAP)
- The Directory System Protocol (DSP)
- The Directory Operational Binding Protocol (DOP)
- The Directory Information Shadowing Protocol (DISP)
- Lightweight Directory Access Protocol (LDAP)

DAP connects a directory application to a DSA and is used by the application to pass directory operations to, and receive results from, the DSA.

DSP connects two DSAs and allows them to cooperate with each other to process directory operations, such as searches.

DOP connects two DSAs and allows them to manage shadowing agreements. A shadowing agreement defines how and when the two DSAs will replicate a given naming context.

DISP connects two DSAs and allows them to replicate information.

The Lightweight Directory Access Protocol (LDAP) provides support for the LDAP V2 and V3 protocols, allowing LDAP clients to access the X.500 directory.

All of these protocols, with the exception of LDAP, are application layer protocols and rely on underlying transport and network connections and the services provided by the OSI Applications Kernel (OSAK). Therefore, any failure of lower layer protocols or connections can result in the loss of the DAP, DSP, DOP, DISP connections. The LDAP protocol runs directly over TCP/IP transport.

DAP, DSP, DOP, DISP connections are established using a Bind request. The transmitting system issues the appropriate bind request, which is then accepted or rejected by the receiving system. The success or failure of DAP, DSP, DOP, and DISP binds can be monitored using four DSA counters:

DUA Binds Accepted
DUA Binds Rejected
Chained Binds Accepted
Chained Binds Rejected
DOP Binds Accepted
DOP Binds Rejected
DISP Binds Accepted
DISP Binds Rejected

Refer to Section 10.2 for more information on these counters. Connection failures are also signalled by events, as described in Chapter 10.

You should monitor counters and events to identify normal system behavior.

This makes it possible to identify abnormal behavior, for example, if the proportion of bind rejects suddenly increases in relation to the number of successful binds.

Table 3.1 shows typical connection problems.

Table 3.1. Connection Problems

Problem	Refer to
Applications Cannot Bind to a DSA	Section 3.1
Connection Between the DUA and the DSA Is Lost	Section 3.2
Response Cannot Be Decoded	Section 3.3
You Receive a ROSE Error	Section 3.4
Event Specifies a Communications Problem	Section 3.5

3.1. Applications Cannot Bind to a DSA

To gain access to the directory, a DUA establishes a connection with a DSA by issuing a DAP Bind request to the DSA. The DUA and DSA exchange information to establish a framework in which they can operate.

Once a connection is established, the DUA can send directory requests to the DSA.

How, when, and to which DSA a connection is established, depends on the DUA issuing the DAP Bind request.

- By default, the DXIM command line interface connects to the address specified in the **DUA.KnownDSAs.paddr** parameter in the DUA defaults file.

  The connection to the DSA is established when the DXIM user enters their first request, such as a search.

  You can use the Bind command with the Address argument to override this default address. See the online help for details of the Bind command.

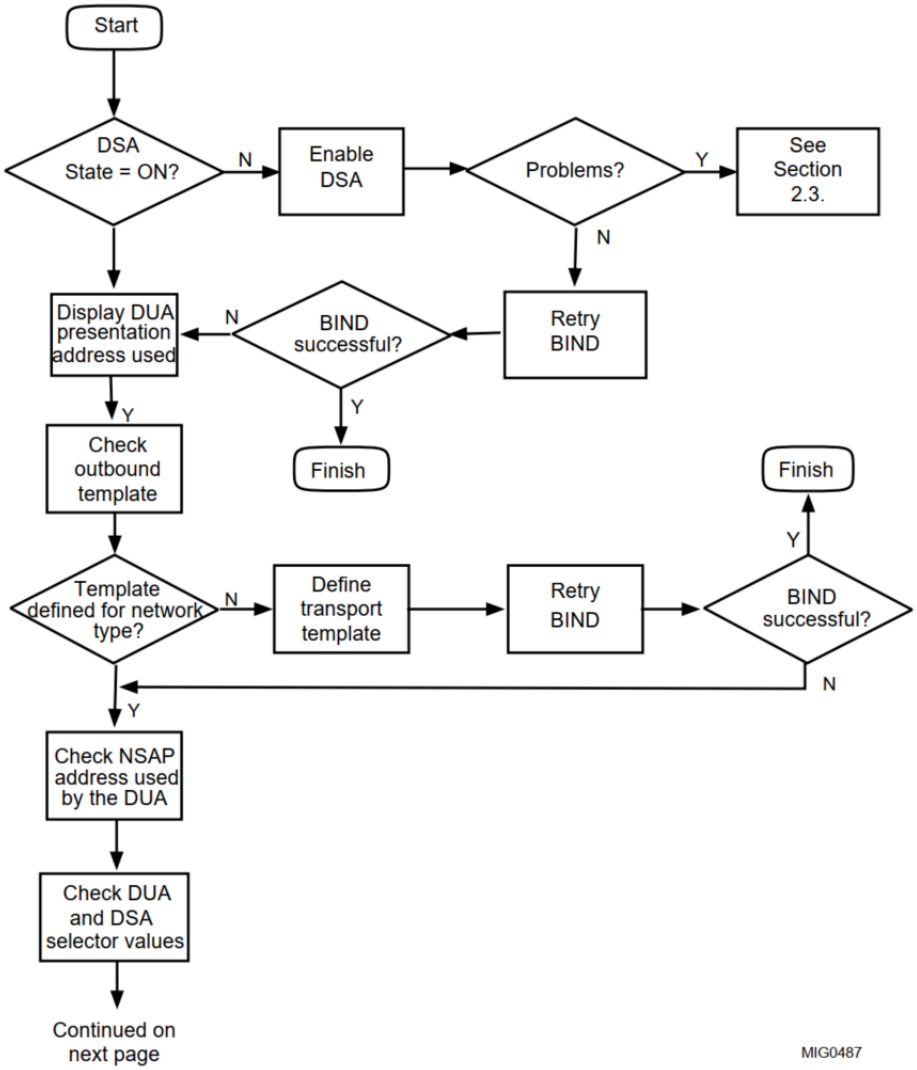
- The DXIM windows interface connects to a DSA when the utility is invoked. It also makes a connection whenever a user authenticates, or if a connection times out between user requests.

  The DXIM windows interface always connects to the address specified in the **DUA.KnownDSAs.paddr** parameter in the DUA defaults file. There is no way to override this.

Any problems with the Bind request are indicated by an error message, for example:

```
Unable to bind to the Directory
```

The following section is written as a sequence of steps to be followed to diagnose and solve a bind problem. Figures below show the steps and Table 3.2 gives an overview of these steps. The solutions described only apply to connections between HP's DSAs and DUAs.


Start
DSA State = ON?
N
Enable DSA
Problems?
Y
See Section 2.3.
N
Display DUA presentation address used
N
BIND successful?
Retry BIND
Y
Y
Check outbound template
Finish
Finish
Y
Template defined for network type?
N
Define transport template
Retry BIND
BIND successful?
N
Y
Check NSAP address used by the DUA
Check DUA and DSA selector values
Continued on next page
MIG0487

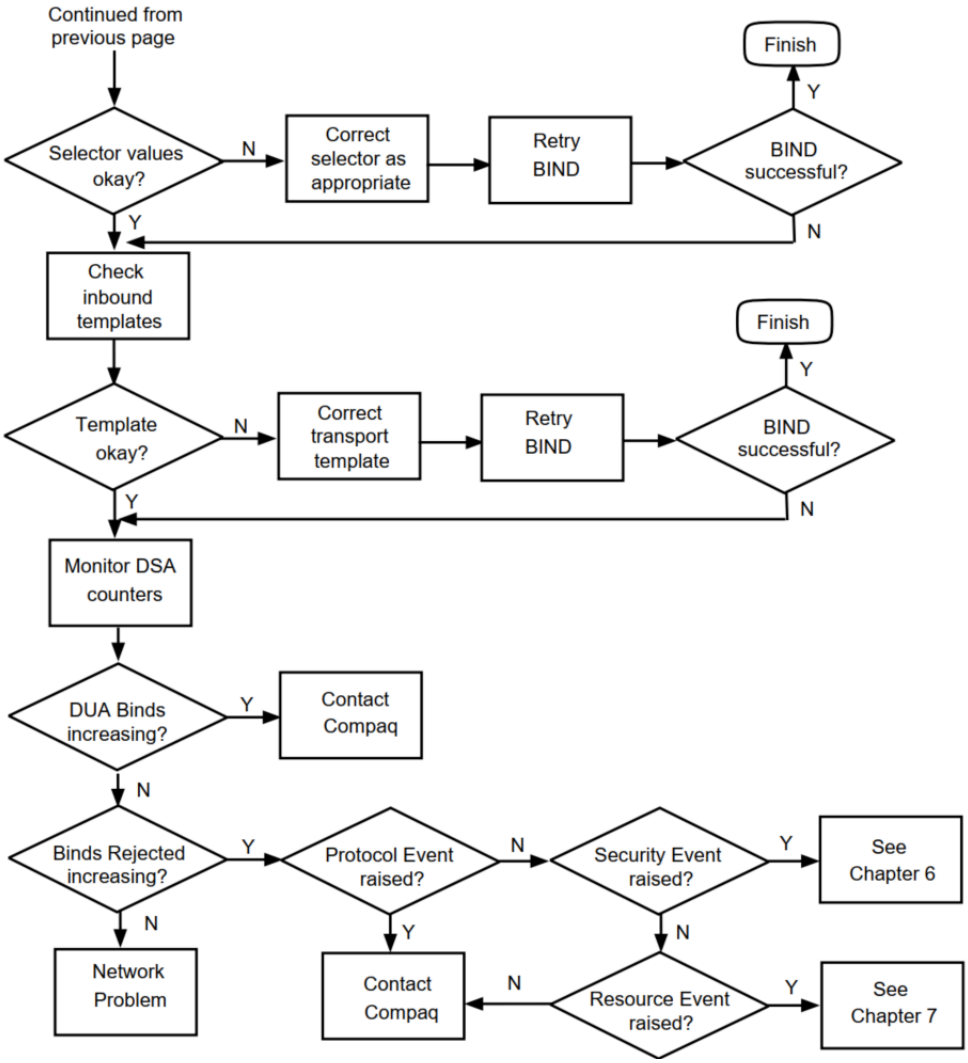


Table 3.2. Problems Binding

Action	Refer to
Check DSA State is ON	Section 3.1.1
Display DUA Presentation Address Used	Section 3.1.2
Check Outbound Template	Section 3.1.3
Check NSAP Address Used by the DUA	Section 3.1.4
Check DUA and DSA Selector Values	Section 3.1.5
Monitor DSA Counters	Section 3.1.6
Investigate the Network Problem	Section 3.1.7
Check for Protocol, Resource and Security Events	Section 3.1.8

3.1.1. Check DSA State is ON

Check that there is a DSA process running on the node to which you are trying to bind, and that the DSA is enabled for communication.

On the node where the DSA is installed, enter the following command to display the state of the DSA:

```
NCL> SHOW DSA STATE
```

The DSA state must be ON.

If the DSA is OFF, use the ENABLE DSA directive.

If the ENABLE directive is successful, try to rebind to the DSA; otherwise, refer to Section 2.4 for information on how to solve problems associated with enabling the DSA.

If you receive a **No Such Entity** exception, it indicates that the DSA is not running. Check this as follows:

Enter the command `$ SHOW SYSTEM`

If the DSA process is running, the DXD$DSA_SERVER process is listed. If the DSA process is not running, start it as follows:

```
$ @SYS$STARTUP:DXD$COMMON_STARTUP
```

3.1.2. Display DUA Presentation Address Used

If your Bind command fails, it may be that the DUA is using an incorrect presentation address in the bind.

The following is a typical example of a DSA presentation address:

```
Presentation Address = '"DSA"/"DSA"/"DSA"/NS+49002aaa00040008aa21,CLNS'
```

Use the NCL SHOW DSA PRESENTATION ADDRESS comman d to check the address of the DSA that you want to connect to. Check that you are specifying that address correctly in your command line, or that it is specified correctly in the `DUA.KnownDSAs.paddr` parameter in the DUA defaults file.

See *VSI Enterprise Directory Management* for details of how to plan and set a DSA's presentation address.

3.1.3. Check Outbound Template

OSI connections to or from a HP DSA depend on some OSI transport templates being defined on the DSA system and the DUA system. Note that RFC1006 connections do not require transport templates. If the DUA was attempting to connect using an RFC1006 address, there is no need to check the templates.

You can run the following procedure to make sure that the necessary templates are defined on the DSA and DUA systems.

```
NCL> DO SYS$STARTUP:DXD$TEMPLATE_STARTUP.NCL
```

Note that the comman ds for creating the DXD_CONS templates are commented out of the startup file. If your system can use connection-oriented networking, edit the startup file to remove the comment characters from the commands that create and configure the DXD_CONS entity, and amend the commands to reflect the configuration of your connection-oriented system. The commands provided are typical examples, but might not match the configuration of your system. See your system manager for details of what values to set for the attributes of the DXD_CONS entity.

After running the procedure on both the DUA and DSA systems, retry the connection.

3.1.4. Check NSAP Address Used by the DUA

Check that the network address specified as part of the DUA's bind address (see Section 3.1.2), correctly identifies a valid network address of the node which holds the DSA. To do so, on the DSA node, display a list of valid NSAPs using the following NCL command:

```
NCL> SHOW OSI TRANSPORT LOCAL NSAP * ALL ATTRIBUTES
```

Ensure that the NSAP address part of the address that the DUA is trying to connect to is one of those displayed by this command.

If the NSAP address being used by the DUA does not match any of the DSA's local NSAPs, amend the NSAP address specified as part of the DUA's presentation address. Do this as follows:

- If you are binding to the DSA using the DXIM command line interface Bind command, then repeat your Bind command using the correct presentation address.
- If the Bind command is within a DXIM script file, amend that script file.
- If DXIM is reading the presentation address from the DUA defaults file, run the DUA configuration procedure to create a new DUA defaults file.

3.1.5. Check DUA and DSA Selector Values

Check the upper layer selectors in the address that DUA is attempting to connect to, as follows:

- On the DSA node, use the following NCL command:

  ```
  NCL> SHOW DSA PRESENTATION ADDRESS
  ```

  Typically, the selectors for a DSA are "DSA"/"DSA"/"DSA".

- Check the upper layer selectors used by the DUA, by looking at the address specified in the DXIM Bind command, or by looking at the `DUA.KnownDSAs.paddr` parameter in the DUA defaults file.

If the values do not match, repeat the DXIM command with the correct address, or run the DUA configuration procedure to create a new DUA defaults file.

3.1.6. Monitor DSA Counters

It might be that the DSA is not receiving the Bind request. This can be checked by monitoring the DSA counters.

Check this as follows:

1. On the DSA node, use the following NCL command:

   ```
   NCL> SHOW DSA DUA BINDS ACCEPTED, DUA BINDS REJECTED
   ```

   The DUA BINDS ACCEPTED counter displays the total number of successful DUA Binds received by the DSA since it was enabled. The DUA BINDS RE J ECTED shows the total number of DUA Binds rejected by the DSA since it was enabled.

2. Reissue your Bind request.

3. Re-examine the counters as in step 1.

If either counter has increased, then the DUA is using a valid address for the DSA, but the DSA is not accepting the connection. See Section 3.1.8 for a list of DSA events to check. The events may indicate the reason why the Bind request was rejected.

If neither counter increased, then it indicates that the DSA is not receiving the Bind request at all and that there is a network problem. See the DECnet-Plus Problem Solving documentation for information on solving network problems.

3.1.7. Investigate the Network Problem

If you have a network problem that is preventing connections to a DSA, you can use the network isolation tool to investigate the problem. Section 3.6 documents the network isolation tool.

3.1.8. Check for Protocol, Resource and Security Events

If the DSA is rejecting your Bind request, as indicated by the DUA BINDS RE J ECTED counter increasing, then check for the following DSA events (see Section 1.3 for information on how to log events):

- Distributed Operation Failure event

  If there is an event containing a ROSE reject status, refer to Section 3.4. A ROSE reject usually indicates an interworking problem with another vendor's implementation.

- Authentication Failure event

  This indicates that the Bind request had insufficient authentication parameters, or authentication parameters that are incorrect. For example, the DSA might require all connections to include a name and password, and might only allow access to users with particular names. Refer to Section 6.4.

- Resource Exhausted event

  This indicates that the DSA has insufficient resources to process the connection request. Refer to Chapter 7.

3.2. Connection Between the DUA and the DSA Is Lost

The loss of the connection between the DUA and a DSA is indicated by a Communications Error message. For an explanation of these error messages, see Chapter 9.

If the connection fails while you are using the DXIM command line interface, retry the command. This forces DXIM to rebind to the DSA. It may be that the loss of the connection was a temporary problem and that a repeat bind may solve the problem. If the command fails repeatedly, it is possible that someone has disabled the DSA.

If the connection fails while you are using the DXIM windows interface, DXIM attempts to re-establish the connection. If the connection cannot be re-established, see Section 3.1.

Typical reasons for a connection being lost may be because the link timed out, or the server or client system has run out of resources, or the DSA has been disabled. Therefore, check to see whether any

events were issued at the server, as a result of the connection being lost. Specifically, check for Resource Exhausted events. If a Resource Exhausted event was issued, see Chapter 7.

3.3. Response Cannot Be Decoded

If DXIM cannot decode a response it issues the following error:

```
Communications Error: DXIM is Unable to Decode the Response From the DSA
```

This means that the DSA is sending DXIM some protocol that DXIM does not understand. This might be because the DSA implements some protocol extensions that DXIM does not support.

If this error occurs, enable OSAK tracing on the DUA node and then repeat the request that caused the error. You then need to decode the OSAK trace to see what is wrong with the protocol, or how the protocol differ s from other requests that DXIM has accepted and decoded successfully. If the problem happens frequently, and you do not know how to decode the protocol, contact VSI, or the vendor of the DSA that generated the protocol.

3.4. You Receive a ROSE Error

ROSE allows the exchange of requests and responses between a DUA and a DSA, or between two DSAs. Generally, a ROSE error indicates an interworking problem between a HP DUA or DSA and another vendor's DUA or DSA.

If you are connected to a HP DSA when you receive the ROSE error, it means that there is an internal software problem. This can be caused by the following things:

- A DUA or DSA is sending invalid protocol data units (PDUs) which the HP DSA does not understand
- A DSA is misinterpreting PDUs received
- A PDU contains an incorrect value encoding

If you receive a ROSE error, rebind to the same DSA and trace the connection using the OSAK trace utility as described in Section 1.5. This will help to determine whether the DUA is at fault by issuing faulty protocol, or whether the DSA is at fault by misinterpreting correct protocol.

If you are unable to correct the error, then contact VSI.

3.5. Event Specifies a Communications Problem

Several DSA events can specify that the reason for a connection failure is a communications problem. The following sections describe how to respond to each of the possible communications problems that can appear in these events.

Table 3.3 lists the possible communications problems:

Table 3.3. Communications Problems

Action	Refer to
Fatal Interface Error	Section 3.5.1

Action	Refer to
Insufficient Resources	Section 3.5.2
Network Unavailable	Section 3.5.3
Address Already in Use	Section 3.5.4
Invalid AEI	Section 3.5.5
Transport Error	Section 3.5.6
System Error	Section 3.5.7
Invalid Transport Template	Section 3.5.8
Unknown Error	Section 3.5.9
ACSE User Reject	Section 3.5.10

3.5.1. Fatal Interface Error

A Fatal Interface Error must be reported to VSI.

3.5.2. Insufficient Resources

Chapter 7 provides details of how to handle resource problems. Check the event log for other DSA events to see whether they provide additional information about the resource problem.

3.5.3. Network Unavailable

This should be a temporary problem. If the error happens frequently, check the event log to see whether any other events indicate why the network is unavailable. There should be some DECnet events that provide further information. This problem needs to be reported to your DECnet or network manager.

3.5.4. Address Already in Use

The presentation address of the DSA is already being used by some other application. Use the NCL SHOW DSA PRESENTATION ADDRESS comman d to check the presentation address, and to verify that you have specified it correctly.

If you find that the address you want to use really does belong to some other application, then you need to choose a different address for your DSA. In this case, you also need to amend any attributes of other DSAs that include this DSA's address. For example, other DSAs might have Subordinate Reference entities that refer to this DSA. If the address specified in those Subordinate References is incorrect, those DSAs will generate errors, and the Enterprise Directory will not work correctly.

3.5.5. Invalid AEI

This problem can be reported for several reasons, and it is difficult to determine which. Check the event log for other events that occurred shortly before or after the DSA reported this problem.

The DSA event that contains this problem should specify the network address that the DSA was attempting to connect to. Check that this address is correct. If not, you need to amend the address.

If the DSA was attempting a DISP operation or DOP operation, then a Consumer Access Point needs to be amended. The event should specify the name of the Naming Context that was being replicated.

Amend the Consumer Access Point of that Naming Context to specify the correct address for the consumer DSA.

If the DSA was attempting to chain a user request, then you need to amend the knowledge reference that the DSA was using. This is usually a Subordinate Reference, but might be a Superior Reference. In order to determine which entity you need to amend, you need to know what entry the DSA was attempting to chain a request for. The user should provide this information.

If the event that contains the problem is a Listen Failure, then you need to check the Presentation Address and AE Title attributes of the DSA entity.

If the error is displayed when you use the UPDATE DSA command, then you need to check the presentation address you specified in the command. If you specified an AE title in the command, then you need to check the presentation address that is contained in the directory entry with the same name as the AE title you specified.

Although these configuration errors might explain an Invalid AEI problem, there are also many other explanations, such as OSAK problems. You need to work with your DECnet manager to determine the real cause of this problem.

3.5.6. Transport Error

Check your DECnet-Plus problem solving documentation for details of how to deal with problems in the transport layer. Check the event log to see whether other applications are also having transport problems.

3.5.7. System Error

Report this problem to VSI.

3.5.8. Invalid Transport Template

The Enterprise Directory requires some OSI transport templates.

On OpenVMS systems, these templates are defined during the Enterprise Directory startup. Shutdown the Enterprise Directory, and restart it to redefine the templates.

3.5.9. Unknown Error

Report this problem to VSI.

3.5.10. ACSE User Reject

Check the event log for other events that indicate communications problems for the local system. Refer to the DECnet-Plus problem solving documentation for details of how to handle ACSE user rejections.

3.6. Testing Network Connections

You can test a network connection using the network isolation tool. The network isolation tool enables you to test the network connection between two applications that are either installed on the same node or on different nodes. For example, you can use the network isolation tool to test the network connection between the following:

- A DSA and an application, such as DXIM.
- Two DSAs

You can use the network isolation tool to test either a DECnet-Plus connection or an RFC1006 connection.

The network isolation tool comprises two components, a server and a client. The client represents the application that initiates the connection and the server represents the application that receives the connection.

For example, to test a connection between an application on one node and a DSA on another node, invoke the server on the node where the DSA is installed. Invoke the client on the node where the application is installed. The two nodes may be running different operating systems. In the case where both applications are installed on the same node, then run the server and client on the same node.

The client and server produce output that specifies whether the connection was successful. The output from both the server and the client identifies any errors that cause the network connection to fail. All error messages in the output refer to OSAK errors. Refer to the DECnet-Plus documentation set for information about OSAK errors.

You can select two levels of output from both the client and server; brief or full. The brief option reports the success or failure of the network connection. The full option reports each stage of the connection. You can specify a different level of output for each of the components.

Use the brief option to test that a connection can be established between the client and the server. If the connection attempt fails, then rerun the tool and specify the full option in order to obtain more information about the problem that caused the connection to fail.

3.6.1. Running the Network Isolation Tool

When you invoke both the client and the server, you are asked a series of questions. If you require help about any of the questions asked by either the client or server, then type HELP or a question mark (?) at the prompt. If you type HELP, then the complete online help for the network isolation tool is displayed. If you enter a question mark (?), then only the help relevant to the question is displayed.

The commands to invoke the client and server have two optional qualifiers, -l and -h. Use the -l qualifier to write the output to a file, you also need to specify the name of the file for the output. If you do not specify the -l qualifier, then the output is displayed on your screen. Use the -h qualifier to display the online help for the network isolation tool. Note that you can use only one qualifier in the command.

By default, the server process exits after processing a client connection. However, you can set up the server process so that it remains active after a connection attempt. This means that you can repeatedly rerun the client and test other connections to the server. The client and server can test connections using only one transport service at a time, for example, DECnet-Plus. If you want to test a DECnet-Plus connection and then an RFC1006 connection, you must rerun the server in order to change transport services.

The following sections explain how to use the server (see Section 3.6.1.1) and the client (see Section 3.6.1.2).

3.6.1.1. Running the Server

Run the server on the node where the application that receives the connection is installed. Use the following command to run the server:

```
$ RUN SYS$ TEST:DXD$NIT_SERVER.EXE
```

In order to use the -l or -h qualifiers, you must specify this command as a foreign command. To do this, define a logical name for the command, for example:

```
$ NIT-SERVER := $SYS$TEST:DXD$NIT_SERVER.EXE
```

Use the following command to specify a qualifier:

```
$ NIT-SERVER [-h] [-l filename]
```

where *filename* is the name of the file for the output.

The server asks the following questions:

1. `Do you want the Server to accept more than one connection [N]:`

   By default, the server process exits after processing a client connection. If you want the server process to remain active after you have run the client, so that you can make another attempt to connect to the server, enter YES; otherwise, press Return.

2. `Will RFC1006 be used for the connection? [N]:`

   If you want to test an RFC1006 connection, enter YES; otherwise, press Return.

3. `Level of output (brief/full) [brief]:`

   Enter FULL for a record of each stage of the network connection; otherwise, press Return.

The server process starts and waits for a connection request from the client.

If you specified the -l qualifier when you invoked the server, then the output from the server is written to the file that you specified. The server displays the following message on your screen:

```
Test output is being written to the log file...
```

If you did not specify the -l qualifier, then the output from the server is displayed on your screen.

When the connection attempt is complete, the server displays the following message on your screen:

```
Done.
```

Each line of the output from the server is identified by the word `Server`.

If you answered YES to the question: `Do you want the Server to accept more than one connection?`, then the server process remains active. After you have completed all your testing you need to manually force the server process to exit. To do this, issue the CTRL/C command on the node where the server is running.

3.6.1.2. Running the Client

Before you run the client, you need to find out the network address (NSAP) for the node where you are running the server. If possible, find out the network address that the application represented by the client uses to establish a connection to the node where you are running the server. If you are using the network isolation tool to find out why a connection failed, find out what NSAP was used in that failed connection.

When the server is running, invoke the client on the node where the application that initiates the connection is installed. If this is the node where you are currently running the server, then you need to create another session on that node in order to run the client.

Use the following command to invoke the client. Note that you can use the same optional -l or -h qualifiers that are available for the command to invoke the server. See Section 3.6.1 for a description of the -l and -h qualifiers.

```
$ RUN SYS$TEST:DXD$NIT_CLIENT.EXE
```

In order to use either of the qualifiers, you must specify this command as a foreign command, as described in the command example in Section 3.6.1.1.

The client asks the following questions:

1. `NSAP address:`

   Enter an NSAP address for the node where you are running the server. Include the network service identifier (NS+) and the network type, for example: `NS+4900ca08002b30254221,CLNS`

   Note that if, when you invoked the server, you answered `YES` to the question: `Will RFC1006 be used for the connection? [N]:`, then you must specify an RFC1006 NSAP.

2. `Level of output (brief/full) [brief]:`

   Enter `FULL` for a record of each stage of the network connection; otherwise, press Return.

The client initiates a connection to the server. If you specified the -l qualifier when you invoked the client, then the output from the client is written to the file that you specified. In this case, the client displays the following message on your screen:

```
Test output is being written to the log file...
```

When the client connection attempt is complete, the client displays the following message on your screen:

```
Done.
```

If you did not specify the -l qualifier, then the output from the client is displayed on your screen.

The output from the client is prefixed with the following information:

- The type of network connection, for example CLNS.
- The OSI transport template being used to make the connection to the server.
- The level of output that you requested.

Each line of output from the client is prefixed with the word `Client`.

If, when you invoked the server, you answered `YES` to the question: `Do you want the Server to accept more than one connection?`, you can rerun the client and test another connection to the server. After you have completed all your test connections to the server, force the server process to exit by issuing the `CTRL/C` command on the node where the server is running.

If you are unable to solve a network connection problem identified by the network isolation tool, then contact VSI about the problem.

3.7. Checking Lightweight Directory Access Protocol

Check that the DSA LDAP port is non-zero by entering the command:

```
NCL> SHOW DSA LDAP PORT
```

Check that the DSA is enabled (See Section 3.1.1).

If the error "No Resource Available" occurs on enable, check that the other application is not using the DSA's LDAP port.

Chapter 4. Problems With Distributed Operations

This chapter covers problems associated with the configuration of a distributed and replicated Enterprise Directory.

VSI Enterprise Directory uses the following knowledge information to perform distributed operations:

- Superior Reference entities
- Subordinate Reference entities
- The Consumer Access Point attribute of Naming Context entities

  The creation of Consumer Access Point attributes causes the automatic creation of shadowing agreement subentries. These subentries can be managed to customize replication.

Table 4.1 shows some problems associated with the configuration of distributed operations.

Table 4.1. Problems with Distributed Operations

Problem	Refer to
Cannot Create a Naming Context	Section 4.1
Cannot Create a Subordinate Reference	Section 4.2
Cannot Delete a Naming Context	Section 4.3
Cannot Delete a Subordinate Reference	Section 4.4
Replication Fails	Section 4.5
Shadowing Agreement Automatic Management Fails	Section 4.6

4.1. Cannot Create a Naming Context

This may be due to any of the reasons shown in Table 4.2.

Table 4.2. Problems Creating a Naming Context

Symptom	Refer to
Cannot Create a Naming Context called "/"	Section 4.1.1
Specified Name has Subordinates	Section 4.1.2
Naming Context Already Exists	Section 4.1.3
Superior Master Naming Context needs a Subordinate Reference	Section 4.1.4
Superior Shadow Naming Context needs a Subordinate Reference	Section 4.1.5
Specified Name is an Entry	Section 4.1.6
Specified Name is an Alias Entry	Section 4.1.7
Identifier is Incorrect	Section 4.1.8

Symptom	Refer to
Alias Entry Prevents Creation	Section 4.1.9

4.1.1. Cannot Create a Naming Context called "/"

VSI Enterprise Directory does not allow you to create a naming context directly at the root of a DSA (/). A naming context must have a distinguished name that contains at least one relative distinguished name.

Refer to *VSI Enterprise Directory Management* for more information on how to plan your naming contexts.

4.1.2. Specified Name has Subordinates

You cannot create a naming context at this position in the DIT because the name you have specified has subordinates. You must create naming contexts in a hierarchical order (top down). Refer to *VSI Enterprise Directory Management* for more information. You have a choice to solutions to this problem:

- Create the new naming context on a DSA that does not already hold subordinate information.

  If you do this, make sure that you also create the required subordinate references.

- Temporarily remove the subordinate information so that you can create the new naming context.

Of these, the first choice is the easiest to manage. However, if the new naming context must be created on a DSA that already holds subordinate information, you can proceed as described below.

If the existing subordinates are master copies of information, you need to delete them temporarily, create the new naming context, and then recreate the deleted subordinates, as follows:

1. Use the DXIM Multiple Entry Management facilities to select the subordinate entries that need to be temporarily deleted, and DXIM Reading and Writing Files facilities to write those entries to a script file.

   See the DXIM online help for further details of these facilities.

2. When you are confident that you have written all subordinates entries to a DXIM script file successfully, use the DXIM Multiple Entry Management facilities to delete the subordinate entries from the DIT.

3. Use NCL to delete the Naming Context entity or entities that contained the subordinate entries.

4. Use NCL to create the new Naming Context entity.

5. Use NCL to create a Subordinate Reference entity or entities to refer to the subordinate naming contexts that you need to recreate.

6. Use NCL to recreate the Naming Context entities you deleted above.

7. Use the DXIM script file to recreate the entries you deleted above.

If the subordinates are shadow copies, they can be more easily deleted while you create the new naming context, as follows:

1. Configure the supplier DSA(s) of the relevant naming context(s) so that this DSA is no longer listed as a consumer.

This causes the shadow naming context(s) to be deleted within a few minutes.

2. After a few minutes, use the NCL SHOW DSA NAMING CONTEXT * command to check that the naming contexts have been deleted. When they have been deleted, proceed to the next step.

3. Create the Naming Context entity whose creation was previously prevented by the presence of the shadow information.

4. Create a Subordinate Reference entities or entities for the subordinate naming context s held on other DSAs.

5. Reconfigure the supplier DSA(s) of the shadow information so that this DSA is once again a consumer of the relevant naming context(s).

If the existing subordinates are a mixture of master and shadow copies, then you need to apply a combination of the above tasks.

In all cases, take care not to create naming contexts that overlap with naming contexts held on other DSAs. It is important that all DSAs agree about where one naming context ends and another begins, otherwise distributed operations may not work as expected, and errors will occur. The creation of the required subordinate references is therefore essential to the maintenance of the distributed DIT. If possible, all naming contexts and subordinate references should be created in their hierarchical order, top down, so that you never need to undo parts of your configuration as described in this section.

4.1.3. Naming Context Already Exists

A naming context already exists at this position in the DIT. You cannot create a naming context where one already exists. You can view this naming context by starting NCL and entering the following command:

```
NCL> SHOW DSA NAMING CONTEXT identifier ALL ATTRIBUTES
```

where *identifier* is the distinguished name of the naming context you were trying to create, and must be enclosed in quotation marks.

4.1.4. Superior Master Naming Context needs a Subordinate Reference

The DSA refused to create the naming context because a superior master naming context has not been terminated. A naming context must be terminated by a Subordinate Reference entity before you can create another Naming Context entity beneath it.

Make sure that the naming context you tried to create is consistent with the plan of your Enterprise Directory. See *VSI Enterprise Directory Management* for information about planning naming contexts.

To create the proposed naming context, perform the following:

1. Terminate the existing naming context by creating a subordinate reference as follows:

   ```
   NCL> CREATE DSA SUBORDINATE REFERENCE name
   ```

   where *name* is the same distinguished name as you used in your original CREATE DSA NAMING CONTEXT directive.

2. Reissue your original command.

   The command should now succeed as the immediate superior naming context is now terminated by the subordinate reference you have created.

4.1.5. Superior Shadow Naming Context Needs a Subordinate Reference

The DSA refused to create the naming context because a superior shadow naming context has not been terminated. A naming context must be terminated by a Subordinate Reference entity before you can create another Naming Context entity beneath it.

To create the proposed naming context, you must first terminate its superior shadow naming context.

To create the naming context, perform the following:

1. Identify the shadow naming context by starting NCL and entering the following command:

   ```
   NCL> SHOW DSA NAMING CONTEXT * ALL ATTRIBUTES
   ```

   The naming context with the distinguished name that is superior to the distinguished name you specified as part of your CREATE DSA NAMING CONTEXT command, is the shadow naming context.

   The Master Access Point attribute of this naming context identifies the DSA that owns this naming context (that is, the master DSA).

2. Go to the master DSA and terminate the existing naming context by creating a subordinate reference as follows:

   ```
   NCL> CREATE DSA SUBORDINATE REFERENCE name
   ```

   where *name* is the same distinguished name as you used in your original CREATE DSA NAMING CONTEXT directive.

3. Return to your local DSA and reissue the UPDATE directive.

4. Reissue your original command.

   The command should now succeed as the immediate superior naming context is now terminated by the subordinate reference you created on the master DSA.

4.1.6. Specified Name is an Entry

An entry already exists with the same name. You cannot create a naming context where a directory entry already exists.

If you wish to create the naming context, you must first delete the existing directory entry.

4.1.7. Specified Name is an Alias Entry

An alias entry already exists with the same name. You cannot create a naming context where an alias entry already exists.

If you wish to create the naming context, you must first delete the existing alias entry.

4.1.8. Identifier is Incorrect

The name of the Naming Context entity you tried to create is incorrect.

Check this name. The name must be in the form of a distinguished name, for example, "/c=US/o=Abacus/ou=Sales/ou=Accounts".

For more information, refer to *VSI Enterprise Directory Management* or the NCL online help.

4.1.9. Alias Entry Prevents Creation

The distinguished name of the naming context contains an alias and this is not allowed. The alias name may be a portion of the name, or the complete name. The name of the alias entry is returned.

To create the naming context, you must replace the distinguished name of the alias entry with the distinguished name of the entry the alias is pointing to (the target entry) and then reissue your command, using the new distinguished name. Do this as follows:

1. Start the DXIM command line interface and enter the following command:

   ```
   dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIAS
   ```

   where *name* is the name of the alias entry, returned in the error message.

   The `Aliased Object Name` attribute specifies the distinguished name of the target entry.

2. Start NCL and reissue your original command, but this time, the distinguished name of the naming context should contain the distinguished name of the target entry and not the distinguished name of the alias.

4.2. Cannot Create a Subordinate Reference

This may be due to any of the reasons shown in Table 4.3.

Table 4.3. Problems Creating a Subordinate Reference

Symptom	Refer to
Subordinate Reference Already Exists	Section 4.2.1
Specified Name is a Naming Context	Section 4.2.2
Cannot Create a Subordinate Reference on a Shadow Naming Context	Section 4.2.3
Cannot Create a Subordinate Reference Called "/"	Section 4.2.4
Specified Name is an Entry	Section 4.2.5
Specified Name is an Alias Entry	Section 4.2.6
Specified Name has Subordinates	Section 4.2.7
Existing Subordinate Reference Prevents Creation	Section 4.2.8
Alias Entry Prevents Creation	Section 4.2.9

Symptom	Refer to
Identifier is Incorrect	Section 4.2.10

4.2.1. Subordinate Reference Already Exists

A subordinate reference already exists at this location in the DIT. Check this by starting NCL and entering the following command:

```
NCL> SHOW DSA SUBORDINATE REFERENCE name ALL ATTRIBUTES
```

where *name* is the distinguished name of the subordinate reference you tried to create. If the entity exists and has the correct attributes, no action is necessary.

4.2.2. Specified Name is a Naming Context

A naming context already exists at this location in the DIT. A subordinate reference cannot be created if a naming context already exists at the location. A subordinate reference is used to terminate a naming context.

4.2.3. Cannot Create a Subordinate Reference on a Shadow Naming Context

You cannot create a subordinate reference on a shadow entry or with a parent that is a shadow.

You must create the subordinate reference on the master DSA and then replicate that particular naming context. Do this as follows:

1. Locate the master naming context by starting NCL and entering the following command:

   ```
   NCL> SHOW DSA NAMING CONTEXT * MASTER ACCESS POINT
   ```

   The naming context with the distinguished name that is superior to the distinguished name you specified as part of your CREATE DSA SUBORDINATE REFERENCE comman d, is the shadowed naming context.

   The Master Access Point attribute of this naming context identifies the DSA that owns this naming context (that is, the master DSA).

2. Go to the master DSA and reissue your original command.

3. Return to your local DSA and reissue the UPDATE directive.

4.2.4. Cannot Create a Subordinate Reference Called "/"

VSI Enterprise Directory does not allow you to create a subordinate reference directly on the root of a DSA (/). A subordinate reference must have a name that contains at least one relative distinguished name.

Refer to *VSI Enterprise Directory Management* for more information on how to plan subordinate references.

4.2.5. Specified Name is an Entry

An entry already exists with the same name. Therefore, there is no need for a subordinate reference of this name.

4.2.6. Specified Name is an Alias Entry

An alias entry already exists with the same name. Therefore, there is no need for a subordinate reference of this name.

4.2.7. Specified Name has Subordinates

Subordinate entries exist below the proposed location of the subordinate reference. To create a subordinate reference in the proposed position would leave these entries beyond the termination point of the naming context.

To create the subordinate reference, you must first delete the subordinate entries.

4.2.8. Existing Subordinate Reference Prevents Creation

A subordinate reference exists further up the DIT on the same naming path from the root, and is the closest hierarchical superior to the reference you are trying to create. The DSA does not allow subordinate references to be created in such a position.

The purpose of a subordinate reference is to mark the termination of a naming context. If the DSA detects that the closest hierarchically superior entity to the subordinate reference you are trying to create is another subordinate reference, then this error is displayed. The normal hierarchical superior to a subordinate reference is a naming context.

It is possible, though unusual, for a subordinate reference to be created with no hierarchically superior entity. You might choose to create such a subordinate reference to provide knowledge of a naming context that does not connect to any part of the DIT held by this DSA. This allows directory requests to access parts of the DIT that do not connect together structurally.

4.2.9. Alias Entry Prevents Creation

The distinguished name of the subordinate reference contains an alias, and this is not allowed. The alias entry is returned.

To create the subordinate reference, you must replace the distinguished name of the alias entry with the distinguished name of the entry the alias is pointing to (the target entry) and then reissue your command, using the new distinguished name. Do this as follows:

1. Start the DXIM command line interface and enter the following command:

   ```
   dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIAS
   ```

   where *name* is the name of the alias entry, returned in the error message.

   The `Aliased Object Name` attribute specifies the distinguished name of the target entry.

2. Start NCL and reissue your original command, but this time, the distinguished name of the subordinate reference should contain the distinguished name of the target entry and not the distinguished name of the alias.

4.2.10. Identifier is Incorrect

The name of the Subordinate Reference entity you tried to create is incorrect. Check this name. The name must be in the form of a distinguished name, for example, "/C=US/O=Abacu s/OU=Sales/CN=Accounts".

A distinguished name must conform to the following rules:

- It must be specified using distinguished name syntax, for example, `/c=US/o=Abacus/cn=DSA2`
- It must not contain an incorrect attribute name keyword, for example, `/county=utah`
- The attribute values must be of the correct syntax. For example, `/c=UK` is not valid because "UK" is not a valid country code.
- You must use printable string characters. Valid characters are:
  - '( )+:-./=?
  - The space character
  - 0123456789
  - abcdefghijklmnopqrstuvwxyz
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ

  Use quotation marks to enclose a string that contains any of the following characters:

  A space character
  comma (,)
  equals sign (=)
  question mark (?)
  slash (/)
  quotation mark (", ‘, or ')

  For example, the following distinguished name contains several RDNs, each of which must be quoted because of the characters within them:

  ```
  /o="Smith, Jones, and Brown Inc."/ou="Sales/Mktg"/cn="Dan 'Swifty' DSA"
  ```

For more information on distinguished names, refer to *VSI Enterprise Directory Management* or the NCL online help.

4.3. Cannot Delete a Naming Context

This may be due to any of the reasons shown in Table 4.4.

Table 4.4. Problems Deleting a Naming Context

Symptom	Refer to
No Such Entity	Section 4.3.1

Symptom	Refer to
Naming Context has Subordinates	Section 4.3.2
Cannot Delete a Shadow Naming Context	Section 4.3.3
Cannot Delete a Naming Context that Contains an Entry	Section 4.3.4
Cannot Delete a Naming Context that Contains an Alias Entry	Section 4.3.5
Alias Entry Prevents Deletion	Section 4.3.6

Note that you should only delete a Naming Context entity if you are changing your Enterprise Directory configuration, or as a temporary measure. See *VSI Enterprise Directory Management* for information about configuring an Enterprise Directory.

4.3.1. No Such Entity

The naming context does not exist. Check that you have supplied the correct distinguished name. You can display a list of naming contexts held by a DSA by starting NCL and entering the following command:

```
NCL> SHOW DSA NAMING CONTEXT *
```

If the naming context is not displayed, it means that the naming context is not held by this DSA.

4.3.2. Naming Context has Subordinates

You cannot remove a naming context if that naming context has subordinate entries or Subordinate Reference entities. Check that the naming context has subordinates as follows:

Start the DXIM command line interface and bind to the relevant DSA. Enter the following command:

```
dxim> SEARCH name NO CHAINING
```

where *name* is the distinguished name of the naming context you tried to delete. If the entry has subordinate entries, these are displayed. If a Subordinate Reference entity exists, a referral is returned.

If you wish you to continue with the deletion, you must first delete all subordinate entries and any Subordinate Reference entities.

4.3.3. Cannot Delete a Shadow Naming Context

You can use the NCL DELETE command to delete a shadow naming context from a DSA but the preferred method is to remove the details of the consumer DSA from the Consumer Access Point attribute of the supplier DSA. This causes the supplier and consumer to terminate replication formally, and the consumer DSA deletes the shadow information automatically.

4.3.4. Cannot Delete a Naming Context that Contains an Entry

The naming context coexists with a directory entry. You cannot delete the naming context until you have deleted the directory entry.

Delete the directory entry and then reissue your command.

4.3.5. Cannot Delete a Naming Context that Contains an Alias Entry

The naming context coexists with an alias entry. You cannot delete the naming context until you have deleted the alias entry.

Delete the alias entry and then reissue your command.

4.3.6. Alias Entry Prevents Deletion

You have tried to delete a naming context using a distinguished name that contains an alias. The name of the alias entry is returned.

To delete the naming context, you must replace the distinguished name of the alias entry with the distinguished name of the entry the alias is pointing to (the target entry) and then reissue your command, using the new distinguished name. Do this as follows:

1. Start the DXIM command line interface and enter the following command:

   ```
   dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIAS
   ```

   where ***name*** is the name of the alias entry, returned in the error message.

   The `Aliased Object Name` attribute specifies the distinguished name of the target entry.

2. Start NCL and reissue your original command, but this time, the distinguished name of the naming context should contain the distinguished name of the target entry and not the distinguished name of the alias.

4.4. Cannot Delete a Subordinate Reference

This may be due to any of the reasons shown in Table 4.5.

Table 4.5. Problems Deleting a Subordinate Reference

Symptom	Refer to
No Such Entity	Section 4.4.1
Naming Context Prevents Deletion	Section 4.4.2
Shadow Naming Context Prevents Deletion	Section 4.4.3
Cannot Delete a Shadow Subordinate Reference	Section 4.4.4
Specified Name has Subordinates	Section 4.4.5
Alias Entry Prevents Deletion	Section 4.4.6

Note that you should only delete a Subordinate Reference entity if you are changing your Enterprise Directory configuration, or as a temporary measure. See *VSI Enterprise Directory Management* for information about configuring an Enterprise Directory.

4.4.1. No Such Entity

The subordinate reference does not exist. Check that you have supplied the correct distinguished name. You can display a list of subordinate references held by this DSA and their distinguished names by starting NCL and entering the following command:

```
NCL> SHOW DSA SUBORDINATE REFERENCE *
```

If the subordinate reference is not displayed, it means that it is not held by this DSA.

4.4.2. Naming Context Prevents Deletion

You cannot delete the subordinate reference because a naming context coexists with the subordinate reference. To delete the subordinate reference would leave the DIT improperly structured.

To delete the subordinate reference, you must first remove the naming context.

4.4.3. Shadow Naming Context Prevents Deletion

You cannot delete the subordinate reference because a shadow naming context coexists with the subordinate reference.

Before you can delete the subordinate reference, you must remove the shadow naming context. See Section 4.3.3 for information about how to remove a shadow naming context.

4.4.4. Cannot Delete a Shadow Subordinate Reference

The Subordinate Reference entity is a shadow copy that has been created through replication. The DSA does not own this Subordinate Reference entity and therefore cannot delete it.

To delete this Subordinate Reference entity, you must delete the Subordinate Reference entity on the master DSA and then retry the replication process. Do this as follows:

1. Use the information returned with the NCL error to identify the master DSA.
2. At the master DSA, modify the consumer information so that the Subordinate Reference entity is not replicated to this shadow DSA.
3. Update the shadow DSA. This removes the shadow Subordinate Reference entity.

4.4.5. Specified Name has Subordinates

The Subordinate Reference entity has subordinates and therefore cannot be deleted. You must delete the subordinate entries or entities before you delete the Subordinate Reference entity.

4.4.6. Alias Entry Prevents Deletion

You have tried to delete a subordinate reference using a distinguished name that contains an alias. The name of the alias entry is returned.

To delete the subordinate reference, you must replace the distinguished name of the alias entry with the distinguished name of the entry the alias is pointing to (the target entry) and then reissue your command, using the new distinguished name. Do this as follows:

1. Start the DXIM command line interface and enter the following command:

   ```
   dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIAS
   ```

   where ***name*** is the name of the alias entry, returned in the error message.

The **Aliased Object Name** attribute specifies the distinguished name of the target entry.

2. Start NCL and reissue your original command, but this time, the distinguished name of the subordinate reference should contain the distinguished name of the target entry and not the distinguished name of the alias.

4.5. Replication Fails

This could be caused by any of the reasons shown in Table 4.6.

Table 4.6. Problems with Replication

Symptom	Refer to
DSA in Wrong State	Section 4.5.1
Consumer Access Point Not Present	Section 4.5.2
Invalid AE Title of Supplier DSA	Section 4.5.3
Cannot Read Supplier Address	Section 4.5.4
Consumer Not Authenticated	Section 4.5.5
Supplier DSA is Unavailable	Section 4.5.6
Update Incompatible with the DSA	Section 4.5.7
Insufficient Resources	Section 4.5.8
DIT Incompatible	Section 4.5.9
Schema Incompatible	Section 4.5.10
DISP Errors Occur Frequently	Section 4.5.11
Shadowing Agreement Incorrectly Customized	Section 4.5.12

4.5.1. DSA in Wrong State

If you use the UPDATE DSA directive when the DSA is not in state ON, the Wrong State error is displayed. Enable the DSA and repeat the UPDATE DSA directive.

4.5.2. Consumer Access Point Not Present

If replication does not seem to be working, but there are no error events indicating that replication has failed, then it is possible that the supplier DSA is configured incorrectly. In this case, replication can complete without error, but without replicating any information, or only a subset of the information you expect, and the DSA can generate the Shadow Update Complete event.

If you find that all or some of the information you expected to consume is not present on the consumer DSA, then you need to check that the Consumer Access Point attribute has been correctly configured on the relevant Naming Context entity on the supplier DSA.

The supplier DSA only provides naming contexts that list the consumer DSA's access point in the Consumer Access Point attribute. *VSI Enterprise Directory Management* describes how to implement replication. When you have configured the supplier correctly, replication should succeed at the next attempt.

4.5.3. Invalid AE Title of Supplier DSA

The UPDATE DSA directive requires you to specify either the presentation address of the supplier DSA, or its AE title. If you specify the AE title, you must use the correct syntax. You can check the AE Title of the supplier DSA by starting NCL and entering the following command:

```
NCL> SHOW NODE node-name DSA AE TITLE
```

where *node-name* is the name of the node containing the supplier DSA. Specify the AE title exactly as displayed by this NCL command.

4.5.4. Cannot Read Supplier Address

The UPDATE DSA directive requires you to specify either the presentation address of the supplier DSA, or its AE title. If you specify the AE title, the DSA attempts to look up the directory entry of that name, to determine the presentation address. If there is no entry with the same name as the specified AE title, or if the entry is inaccessible, then the UPDATE DSA command fails. In this case, the Cannot Read Supplier Address message is displayed.

To solve this problem, you can either specify a presentation address instead, or make sure that the entry of the relevant name exists, and is accessible to this DSA. VSI recommend that all DSAs are represented by **decDSA** entries with the same name as their AE titles, and that you use replication to ensure that copies of these entries are held by every DSA.

4.5.5. Consumer Not Authenticated

If a supplier DSA receives a replication request from another DSA, but that DSA has not specified a password, then the supplier DSA rejects the request because the Consumer cannot be authenticated. It is a security risk to accept replication requests from a DSA whose identity cannot be verified by means of authentication.

The only reason this can happen is if the consumer DSA does not have a Password characteristic attribute, and replication was initiated using the UPDATE DSA command. The consumer DSA will have generated a Shadow Update Failure with the Unexpected Failure reason. The consumer may also generate some Internal Error events in this case. The Internal Error events can be ignored.

Inform the manager of the consumer DSA that they must use the NCL SET DSA PASSWORD command. The password specified for the DSA must match the password in the `decDSA` entry that represents it. Alternatively, the supplier DSA can have an Accessor entity representing the consumer DSA, although this is not the preferred method. The supplier DSA can then verify the password by reference to the `decDSA` entry or the Accessor entity, such that the Consumer Not Authenticated message never recurs.

If the consumer DSA has a password, but it does not match the password in the `decDSA` entry or the Accessor entity, then the supplier DSA generates an Authentication Failure event rather than a Shadow Update Failure event.

4.5.6. Supplier DSA is Unavailable

Replication may fail due to the supplier DSA being unavailable. This could be due to any of the following:

- Node unavailable

Repeat the UPDATE DSA directive when the remote node is available.

- Supplier DSA in the wrong state

  Enable the supplier DSA. It must be in state ON when replication begins.

- Connection to supplier DSA is broken

  Retry the replication. It may be that the connection was aborted due to a temporary problem. If the problems recurs, see Section 3.2.

- Supplier DSA cannot verify the consumer DSA's password.

  The supplier DSA generates an Authentication Failure event. See Section 6.2 for information about how to solve this problem.

4.5.7. Update Incompatible with the DSA

The information provided by the supplier DSA is incompatible with the information at the consumer DSA. After this has occurred, the consumer DSA is automatically deleted to ensure that the consumer DSA's database is not corrupted by the attempted replication operation.

Replication between the two DSAs is not possible and should not be subsequently attempted until the incompatibility has been resolved.

The supplier DSA may provide incompatible information for one of the following reasons:

- The schema files are different on the supplier and consumer DSAs. See Section 4.5.10 for details of how what to do in this case.

- The information that has been replicated to the consumer DSA clashes with information already present on the consumer.

  See Section 4.5.9 for details of how what to do in this case.

- The supplier and the consumer DSAs are using different, incompatible versions of the DSA software.

  VSI is not aware of any replication incompatibilities between the different versions of the HP DSA. Every version of the DSA has provided backwards compatibility. However, if you find an incompatibility, report it to VSI. Use the SHOW DSA VERSION command to find out what versions of DSA you are using, and report this information to VSI.

4.5.8. Insufficient Resources

Refer to Section 7.3.

4.5.9. DIT Incompatible

This message can appear in Shadow Update Failure events if replication fails between two DSAs. It means that the information received by the con sumer DSA clashes with the information it already holds.

This indicates that there is a fault in your Enterprise Directory configuration.

Use the SHOW DSA NAMING CONTEXT * and SHOW DSA SUBORDINATE REFERENCE * to see what information is already present on the consumer DSA. Check that none of the information that

is to be replicated will clash with information already held. For example, if the replicated naming context is hierarchically subordinate to an existing naming context, check that the existing naming context is correctly terminated by a Subordinate Reference entity.

VSI Enterprise Directory Management describes the restrictions that apply to configuring naming contexts, and explains the need for Subordinate Reference entities to mark the end of a naming context as well as to provide a reference to the immediately subordinate naming context.

4.5.10. Schema Incompatible

This message can appear in Shadow Update Failure events if replication fails between two DSAs. It means that the information received by the consumer DSA depends on schema definitions that are not defined in the consumer DSA's schema.

VSI recommends that all DSAs have the same schema. Find out what schema differences there are between the two DSAs, and make the necessary amendments to ensure that the two sets of schema are aligned. In particular, make sure that the consumer DSA has the schema that describes the entries in the naming context that you want it to consume. See *VSI Enterprise Directory Management* for full details of how to customize the schema.

4.5.11. DISP Errors Occur Frequently

Section 10.1.18 provides a list of DISP errors that can occur during replication. Most of these errors are self correcting. However, if any DISP error occurs frequently, it may be that the DSAs cannot correct the problem automatically.

To resolve this problem, use the UPDATE DSA command on the consumer DSA, specifying its AE title. For example:

```
ncl> UPDATE DSA SUPPLIER "/C=US/O=Abacus/CN=DSA1"
```

If the supplier DSA is not represented by a **decDSA** entry in the Directory, then you have to specify the presentation address of the supplier DSA instead of its AE title. For example:

```
ncl> UPDATE DSA SUPPLIER '"DSA"/"DSA"/"DSA"/RFC1006+mynode,RFC1006'
```

The UPDATE DSA command makes the supplier DSA give the consumer DSA a full update of the information that it is configured to consume. The consumer and supplier DSAs also update their shadowing agreements. This should mean that the DISP errors stop happening.

4.5.12. Shadowing Agreement Incorrectly Customized

With this version of the Enterprise Directory, the documentation explains how you can customize shadowing agreements. For example, you can change the replication schedule, or specify that changes are replicated as soon as they occur, rather than after certain intervals.

It is possible that managers will change a shadowing agreement incorrectly if they do not understand agreements fully, and they do not follow the documented instructions. For example, a manager might specify an inappropriate combination of shadowing flags, such that a consumer DSA believes that it is a supplier DSA, or such that each DSA thinks the other will initiate replication. Such configuration errors might not be resolvable automatically, and replication may fail or malfunction as a result.

If replication of a given naming context stops working, or works incorrectly, after you have customized an agreement, there are two solutions:

- Attempt to fix the relevant shadowing agreement subentries manually

  If you configured the subentries incorrectly once, then attempting to fix them manually might not be advisable. However, if you want to modify the agreements manually, refer to *VSI Enterprise Directory Management*.

- Reconfigure the relevant replication, as follows:

  1. On the supplier DSA, delete the Consumer Access Point attribute value that identifies the relevant consumer DSA.

  2. On the consumer DSA, delete the relevant shadow Naming Context. This may happen automatically as a result of step 1.

  3. On the supplier DSA, add the details of the consumer DSA back to the Consumer Access Point attribute of the Naming Context.

     This causes the supplier DSA to create a new, default shadowing agreement subentry, and to communicate with the consumer DSA to tell it to create the corresponding subentry.

     If the two DSAs trust each other, this all happens automatically, and the consumer DSA receives a new copy of the Naming Context. If this does not happen automatically, use the UPDATE DSA command on the consumer DSA to force replication to occur.

  4. When the replication has completed, you can consider customizing the new, default agreement, as documented in *VSI Enterprise Directory Management*. Read the sections on managing shadowing agreements carefully before making any further customizations. Take particular care when modifying the shadowing flags.

4.6. Shadowing Agreement Automatic Management Fails

DSAs use shadowing agreements to store information about which DSAs they should replicate with. A given shadowing agreement describes the replication of a single naming context to a single DSA.

The DSAs manage these shadowing agreements automatically. You must not attempt to manage the shadowing agreements manually, and should be careful not to delete them. Shadowing agreements are represented as directory entries immediately subordinate to the naming context to which they apply[1]. The name of a shadowing agreement subentry indicates whether the DSA that holds the subentry is the supplier or consumer DSA, and the name of the other DSA to which the agreement applies, and an agreement identifier. For example, the following is a typical distinguished name for a shadowing agreement subentry:

```
/C=US/O=Abacus/CN="consumer /C=US/O=Abacus/CN=DSA1 1"
```

This indicates that this DSA is the consumer of the naming context called `/C=US/O=Abacus`, that the supplier DSA is called `/C=US/O=Abacus/CN=DSA1`, and that the agreement identifier is 1.

The supplier DSA will have a corresponding shadowing agreement subentry, which in this case might be called:

```
/C=US/O=Abacus/CN="supplier /C=US/O=Abacus/CN=DSA2 1"
```

[1] If you use the search operation to look for shadowing agreements, you will usuallynot see them. To find subentries you need to use the `subentries` control. See theonline help for the DXIM SEARCH command for further details.

The two entries represent the same agreement.

If you delete a shadowing agreement subentry, it will probably be recreated automatically by the DSA. However, you should try not to delete them because they describe the terms of the replication agreement between the two DSAs, and the replication schedule will be interrupted at least temporarily.

The two DSAs to which an agreement applies both hold an agreement subentry and either DSA might attempt to amend the agreement. For example, either DSA might attempt to force the agreement to be rescheduled.

When two DSAs communicate to manage their respective copies of an agreement, there are a number of problems that can occur, and these are described in Section 10.1.16. Most of the problems are self correcting. Table 4.7 lists the problems that require manual intervention:

Table 4.7. Problems with Shadow Agreement Management

Symptom	Refer to
Shadowing Agreement Invalid	Section 4.6.1
Shadowing Agreement Currently Not Decidable	Section 4.6.2

4.6.1. Shadowing Agreement Invalid

If a Shadowing Update Agreement Failure event specifies the Invalid Agreement problem, this indicates one of two problems. Both of the problems indicate that your Enterprise Directory is incorrectly configured.

The causes of the Invalid Agreement problem are:

- The proposed consumer DSA is actually the master DSA for the naming context that the agreement applies to. The DSA therefore refuses to accept the agreement.

  The event specifies the name of the naming context to which the agreement applies. Use the SHOW DSA NAMING CONTEXT command to see whether the intended consumer DSA is actually the master DSA for the specified naming context.

  If so, remove that DSA's details from the Consumer Access Point attribute of the naming context on the supplier DSA. It is illogical and invalid to configure a DSA to replicate a naming context to the master DSA for that naming context.

- The proposed consumer DSA holds an entry with the same name as the naming context that the agreement applies to, and therefore refuses to accept the agreement.

  The event specifies the name of the naming context to which the agreement applies. Use the following DXIM command to see whether there is already an entry of the same name on the consumer DSA:

  ```
  dxim> show name no chaining
  ```

  If the command returns the entry, then this indicates that you have at least two DSAs with different understandings of how you have divided your DIT into naming contexts. The supplier DSA believes the entry to be at the root of a naming context, but the consumer DSA believes it to be within a naming context.

  If this is the case, then you have a faulty DIT configuration, and you need to review the configuration of all DSAs. DSAs must not have different understandings of where naming contexts begin and end.

4.6.2. Shadowing Agreement Currently Not Decidable

If a Shadow Agreement Update Failure event specifies the Currently Not Decidable problem, this might require manual intervention. If the problem only occurs infrequently, then you need take no action. However, if the problem occurs frequently, it indicates that there is something wrong with the agreement, and that replication is unlikely to function correctly for that agreement.

If the problem occurs frequently, refer to the event to identify the supplier DSA for this agreement. Both of the DSAs should have created corresponding events, each indicating which is the consumer and which is the supplier.

Remove the details of the consumer DSA from the Consumer Access Point attribute of the relevant naming context on the supplier DSA. The AE title and presentation address of the consumer DSA are listed in the event, as is the name of the naming context. For example, you might use the following command to remove the details of the consumer DSA from the attribute:

```
ncl> REMOVE DSA NAMING CONTEXT name CONSUMER ACCESS POINT -
_ncl> {[AE TITLE=aetitle, PRESENTATION ADDRESS=paddr]}
```

Where *name* is the naming context name, *aetitle* is the AE title of the consumer DSA, and *paddr* is the presentation address of the consumer DSA.

Removing the details of the consumer DSA from the attribute causes the supplier DSA to terminate the agreement. The supplier DSA inform s the consumer DSA that the agreement is terminated, and the consumer DSA deletes its copy of the agreement, and of the relevant naming context. When this has succeeded, you can then add the consumer DSA details back to the Consumer Access Point attribute. This creates a new agreement, and should solve the problem.

If this does not solve the problem, and the new agreement also causes the Currently Not Decidable problem, report the problem to VSI.

Chapter 5. Problems With Data Management

This chapter describes some of the more common problems associated with entering information into the directory and receiving information from the directory and explains how to solve them. All of the problems associated with using the Enterprise Directory are highlighted by error messages returned to the user. If the problem you are trying to solve is not described in this chapter, refer to the error message and recovery information contained in Chapter 9.

This chapter explains the problems listed in Table 5.1.

Table 5.1. Problems Manipulating Directory Information

Problem	Refer to
User Receives Information that Is Out of Date or Wrong	Section 5.1
User Continually Receives Referrals	Section 5.2
Information Known to Exist Cannot Be Retrieved	Section 5.3
Attribute Values are Returned in the Wrong Order	Section 5.4
Problems Compiling the Schema	Section 5.5
Cannot Create an Entry of a Specific Class (Motif Interface Only)	Section 5.6
You Want to Backup the Database While the DSA is Running	Section 5.7

5.1. User Receives Information that Is Out of Date or Wrong

This could be caused by any of the reasons shown in Table 5.2.

Table 5.2. Problems with Accuracy of Information

Symptom	Refer to
User Is Using Copy Entries	Section 5.1.1
Frequency of Replication Too Low	Section 5.1.2
Alias Points to Wrong Entry	Section 5.1.3

5.1.1. User Is Using Copy Entries

Replication of naming contexts within the directory means there is no guarantee that the information you get from the directory is up to date, unless you specify that the Enterprise Directory should only return information from master entries.

By default, the Enterprise Directory returns the requested information from the first instance of the entry it finds, which might be a shadow copy. The DXIM command line interface uses an asterisk (*) in the left margin to mark information that was obtained from a copy of an entry.

By setting the appropriate service control in DXIM, you force the directory to return the requested information from the master copy of the directory entry only. This ensures that you get the most up-to-date information.

If you need to be sure that all information displayed has been obtained from master copies, use the Master Information control, as follows:

```
dxim> SHOW name ALL ATTRIBUTES MASTER INFORMATION
```

where *name* is the distinguished name of an entry.

If you always want master information, you can use the DXIM SET DEFAULT command to specify the control for all subsequent commands.

5.1.2. Frequency of Replication Too Low

If your users consistently receive information that is out of date, try increasing the frequency of updates between your supplier DSAs and your consumer DSAs.

5.1.3. Alias Points to Wrong Entry

If a user receives information that is wrong, it may be that the entry specified is an alias entry that is not defined correctly.

Check whether the entry is an alias or not, by typing the following commands:

```
dxim> SHOW ENTRY name ALL ATTRIBUTES
dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIASES
```

where *name* is the name of the entry.

If both commands return the same response, the name specified is not an alias.

If the last RDN of the name is an alias, the response to the second command includes the `Aliased Object Name` attribute, which contains the distinguished name of the target entry. If any other portion of the name is an alias, the response to the second command includes the matched portion of the name.

If the entry is not an alias, repeat this command for the parent entry by removing the last RDN from the distinguished name. Continue this operation until all RDNs have been checked, or an alias entry has been located.

If an alias entry exists, ensure that the value of the `Aliased Object Name` attribute points to the correct target entry.

5.2. User Continually Receives Referrals

If the DSA to which you are bound cannot process your request, it will attempt to chain that request to another DSA which it thinks can satisfy the request. If, for any reason, the chained DSA cannot be reached, the DSA to which you are currently bound will return a referral and generate a Distributed Access Failure event stating the reason why the remote DSA could not be reached.

If a DSA cannot be reached, it may be due to any of the reasons shown in Table 5.3.

Table 5.3. Problems Reaching a DSA

Symptom	Refer to
DSA Prohibit Chaining Attribute Set	Section 5.2.1

Symptom	Refer to
Insufficient Authentication	Section 5.2.2
Node Unavailable	Section 5.2.3
Chained DSA Is Disabled	Section 5.2.4
Connection to DSA Is Broken	Section 5.2.5

5.2.1. DSA Prohibit Chaining Attribute Set

The DSA characteristic attribute Prohibit Chaining overrides the setting of any service controls specified by the user as part of a DXIM command.

Check the setting of the DSA Prohibit Chaining attribute by running NCL and entering the following command:

```
NCL> SHOW DSA PROHIBIT CHAINING
```

If the attribute is set to TRUE, it indicates that this DSA will not perform chaining. To enable chaining, enter the following commands:

```
NCL> SET DSA PROHIBIT CHAINING FALSE
```

Re-execute the command.

5.2.2. Insufficient Authentication

A DSA may not chain a request to another DSA if it cannot maintain the

user's level of authentication. See *VSI Enterprise Directory Management* for information about the service controls that determine whether a DSA chains a request. If, at any point, the DSA cannot continue to chain the request because of authentication, it will return a referral and generate a Distributed Access Failure event with a reason of DSA Not Trusted. Check for this event.

If this event is being generated, check that the DSAs trust each other (see Section 6.2).

If the DSAs are not intended to trust each other, the user must do one of the following:

- Bind directly to the DSA identified in the referral.
- Retry the command, specifying the service control Prefer Chaining instead of Allow Chaining. The Prefer Chaining service control means that you would prefer the DSA to chain the request, rather than return a referral. This means that the DSA will continue to chain the request, irrespective of authentication.

  Note, however, that although the request is chained to another DSA, the user might not have sufficient access rights for the requested operation. Therefore, an error might still be returned.

5.2.3. Node Unavailable

If the remote DSA cannot be reached, it may be that its node is unavailable. This is indicated by a Distributed Access Failure event being generated with a reason of Association Establishment Failure.

Check that the node is present on the network. See Chapter 3 for information about communications problems.

5.2.4. Chained DSA Is Disabled

This is indicated by a Distributed Access Failure event being generated with a reason of Association Establishment Failure.

Check that the DSA is ready for communication by running NCL and entering the following command:

```
NCL> SHOW NODE node-name DSA STATE
```

where *node-name* is the name of the node you are trying to contact.

The DSA must be in state ON.

5.2.5. Connection to DSA Is Broken

If the DSA is currently unreachable, this is indicated by a Distributed Access Failure event being generated with a reason of Association Establishment Failure.

Use the information contained within the referral and attempt to bind directly to the DSA. If the Bind request fails, refer to Chapter 3.

5.3. Information Known to Exist Cannot Be Retrieved

This could be caused by any of the reasons shown in Table 5.4.

Table 5.4. Problems Retrieving Information

Symptom	Refer to
Insufficient Access Rights	Section 5.3.1
Missing Superior Reference	Section 5.3.2
Missing Subordinate Reference	Section 5.3.3
Invalid Reference	Section 5.3.4
Incomplete Knowledge in First Level DSA	Section 5.3.5
Wrong Setting of Local Scope Service Control	Section 5.3.6
Chaining Prohibited	Section 5.3.7
DSA Cannot Be Reached	Section 5.3.8
Shadow Naming Context Out of Date	Section 5.3.9

5.3.1. Insufficient Access Rights

The most likely reason for being unable to retrieve information from the directory is that the user has insufficient access rights to access that information. Refer to Section 6.1 for information on how to solve access control problems.

5.3.2. Missing Superior Reference

If a DSA cannot satisfy a directory request and has no specific knowledge of where to chain to, it passes the request to its superior DSA, in an attempt to locate the required entry. It does this on the assumption

that a DSA with a naming context located nearer the root may have knowledge information about other parts of the DIT. Note that if you configure your Enterprise Directory as recommended in *VSI Enterprise Directory Management* , no DSA requires a superior reference.

Check that your DSA has a superior reference by running NCL and entering the following command:

```
NCL> SHOW DSA SUPERIOR REFERENCE ALL ATTRIBUTES
```

This displays your DSA's Superior Reference. If a superior reference is undefined, define it using the following command:

```
NCL> CREATE DSA SUPERIOR REFERENCE name ACCESS POINT address
```

where *access-point* is the access point of the superior DSA.

See *VSI Enterprise Directory Management* for information about setting up knowledge information.

5.3.3. Missing Subordinate Reference

A subordinate reference is used to identify a DSA that holds a naming context that is subordinate to the naming contexts held by this DSA.

If a subordinate reference is missing, the DSA will not realize that there is a subordinate naming context, and returns the error message **The entry does not exist** for any of the entries held by that naming context.

Check your subordinate references by running NCL and entering the following command:

```
NCL> SHOW DSA SUBORDINATE REFERENCE ALL ATTRIBUTES
```

5.3.3.1.

This displays a list of all subordinate references on your local DSA. There should be one subordinate reference for every naming context subordinate to this DSA. Check this using the DSA worksheets you created while planning your DIT (see *VSI Enterprise Directory Management*). If a subordinate reference is missing, define one using one of the following commands:

```
NCL> CREATE DSA SUBORDINATE REFERENCE name ACCESS POINT address
NCL> CREATE DSA SUBORDINATE REFERENCE name COPY ACCESS POINT address
```

where *name* is the distinguished name of the subordinate entry and *address* is the access point of the DSA containing that subordinate entry. See *VSI Enterprise Directory Management* for details of how to define a subordinate reference.

5.3.4. Invalid Reference

A knowledge reference used while processing a user request is incorrectly defined. For example, the information in the reference does not refer to a DSA, or it refers to a DSA that does not hold the relevant part of the DIT. If the invalid reference was a Superior Reference, the error can mean that the specified DSA does not hold superior information to the DSA that used the reference. If the reference was a Subordinate Reference, then the error can mean that the specified DSA does not hold the relevant subordinate naming context.

The invalid reference could be held on any of the DSAs that were trying to process the user request, not necessarily the user's local DSA.

An invalid knowledge reference causes the DSA to generate a Distributed Operation Failure event with the reason argument set to Invalid Reference and the following error message being returned to the user:

```
DSA Error: Enterprise Directory is Unable to Perform this Command Invalid
 Reference Supplied
```

Refer to Section 9.2 for a description of this DSA error message and the solution to the problem.

5.3.5. Incomplete Knowledge in First Level DSA

If you have designed your DIT as described in *VSI Enterprise Directory Management*, you should have one DSA that holds an entry representing your organization as a whole. This DSA should have knowledge of all other DSAs in your organization and of the naming contexts held by them.

If the knowledge information held by this DSA is incomplete, some of your DSAs will be unaware of the existence of other DSAs, and of parts of the DIT held by those DSAs.

Check that the DSA that holds the entry representing your organization has knowledge of all the other DSAs in your organization. A DSA should also have subordinate references for all immediately subordinate DSAs.

See *VSI Enterprise Directory Management* for information about configuring a distributed DIT.

5.3.6. Wrong Setting of Local Scope Service Control

The Local Scope service control enables you to limit the parts of the DIT that are used to satisfy a request. If the Local Scope service control is in operation, a DSA will only return information from within the part of the DIT it holds. If it cannot satisfy the request, it returns a message indicating that the information does not exist. The default, and recommended, value for the Local Scope service control is FALSE.

The Local Scope service control is specified in the `DUA.LocalScope` parameter of the DUA defaults file. You can also specify a scope in a DXIM command, to override the value set in the defaults file. See the online help for DXIM for more information.

5.3.7. Chaining Prohibited

If the information the user requests is not held by this DSA, the DSA needs to contact other DSAs in order to satisfy the request. The DSA will return a referral instead in any of the following cases:

- The user has specified that chaining is prohibited.
- The DSA is configured to prohibit chaining.
- The DUA defaults state that chaining is prohibited.
- The user is authenticated to the DSA, the remote DSA is not trusted, and Allow Chaining is the default.

The DSA characteristic attribute Prohibit Chaining overrides the setting of any service controls specified as part of a DXIM command.

Use the NCL SHOW DSA PROHIBIT CHAINING command to check the setting of the DSA Prohibit Chaining attribute. If the attribute is set to TRUE, it indicates that this DSA will not perform chaining. To enable chaining, enter the following commands:

```
NCL> SET DSA PROHIBIT CHAINING FALSE
```

Check whether the Allow Chaining, Prefer Chaining or Chaining Prohibited service controls are defined in the DUA defaults file.

Retry your command.

5.3.8. DSA Cannot Be Reached

Refer to Section 5.2.

5.3.9. Shadow Naming Context Out of Date

The entry that cannot be foun d is present in the master naming context but not in the shadow. See Section 5.1.1.

5.4. Attribute Values are Returned in the Wrong Order

A user or application manager might complain that the values of multi-valued attributes are displayed in the wrong order. They might observe that the order changes from one operation to the next, or that the order they added the values in is not preserved by the Enterprise Directory.

Application managers, in particular, might complain because their application assumes that the order of values is significant. They might expect the first value to have precedence over the second, and so on.

This is not a fault with the Enterprise Directory software. The X.500 recommendations state that values are stored as sets rather than sequences of values. To ignore this would cause interworking problems. Users and applications should be encouraged to be tolerant of this if possible.

However, if there is a real need for one value to be recognizable as the primary value of an attribute, *VSI Enterprise Directory Management* explains how to define primary and secondary attributes. It shows how you might define a `favouriteSport` attribute as well as a `Sport` attribute. This enables a person's favourite sport to be easily distinguishable from the full list of sports that a person enjoys.

The default schema includes examples of this strategy such as the `RFC882 mailbox` and the alternative `mail`.

5.5. Problems Compiling the Schema

During installation, a compiled version of the default schema is installed. Therefore, the following errors only apply if you have amended the default schema and attempted to recompile it.

Schema compilation problems could be caused by any of the reasons shown in Table 5.5.

Table 5.5. Problems Compiling the Schema

Symptom	Refer to
Missing Source Files	Section 5.5.1
Missing Attribute Definitions	Section 5.5.2

Symptom	Refer to
Missing Referenced Object Classes	Section 5.5.3
Missing Referenced Name Forms	Section 5.5.4
Missing Referenced Structure Rules	Section 5.5.5
Matching Rules Not Applicable to Syntax	Section 5.5.6
Duplicate Structure Rule Identifiers Found	Section 5.5.7
Superclass Wrong Kind for Class	Section 5.5.8
Too Many Structural Superclass Chains	Section 5.5.9
Wrong Kind of Object Class for Name Form	Section 5.5.10
Duplicate Keyword	Section 5.5.11
Multiple Windows for Name Form	Section 5.5.12
Cannot Open Input File	Section 5.5.13
Cannot Write Schema Output File	Section 5.5.14
Loop Detected While Processing	Section 5.5.15

5.5.1. Missing Source Files

The schema is installed as a number of text files with the file extensions ".sc". On an OpenVMS system, they are installed in the directory defined by the system logical name DXD$DIRECTORY. The files are called:

dec.sc
dit.sc
dua.sc
DXD$SCHEMA.SC
mts.sc
x400.sc
x500.sc
cosine.sc
quipu.sc
entrust.sc

Check that you have all the necessary source files, including any additional files created specifically for your Enterprise Directory. Check that all the necessary files are included by dxd_schema.sc or DXD $SCHEMA.SC.

5.5.2. Missing Attribute Definitions

Every attribute stored in the Enterprise Directory must be defined in the directory schema. Therefore, if you have defined a new auxiliary class, for example, ensure that each of the attributes belonging to that class is defined in the directory schema. Refer to *VSI Enterprise Directory Management* for more information on defining attributes.

5.5.3. Missing Referenced Object Classes

The directory schema defines the hierarchy for object classes.

Ensure that all referenced object classes in a class definition actually exist.

5.5.4. Missing Referenced Name Forms

Name form s are defined in the directory schema and are referenced by structure rule definitions. If you have created a new structure rule, ensure that the name form you refer to is defined in the schema.

5.5.5. Missing Referenced Structure Rules

The directory schema defines structure rules related to some object classes used within the Enterprise Directory. Structure rules define the relationships between different structural classes. If you have created a new structure rule, ensure that any superior structure rules are defined.

5.5.6. Matching Rules Not Applicable to Syntax

When you define an attribute, you specify which matching rules apply to that attribute's values. It is possible to specify each of equality, ordering, substring, and approximate matching rules. However, the choice of matching rule is constrained by the choice of syntax for the attribute.

If the schema compilation fails, and states that a matching rule is not applicable to the syntax, refer to *VSI Enterprise Directory Management* for details of which matching rules may be applied to which syntaxes. That book contains an appendix which details which matching rules can be used for each syntax for the various types of matching. Choose an appropriate rule, amend your schema, and recompile.

5.5.7. Duplicate Structure Rule Identifiers Found

Every structure rule must have a unique numeric identifier. If you see this error, then you have defined a structure rule with an identifier that is also assigned to some other structure rule.

Change the structure rule identifier and recompile.

5.5.8. Superclass Wrong Kind for Class

When defining a new subclass, you must specify whether the class is structural, auxiliary, alias, or abstract.

- A structural class can only be a subclass of another structural class, or an abstract class.
- An auxiliary class can only be a subclass of another auxiliary class, or an abstract class.
- An alias class can only be a subclass of another alias class, or an abstract class.
- An abstract class can only be a subclass of another abstract class.

If you see this error, then you have defined a new class to be a subclass of the wrong type of class. For example, you might have specified that a new alias definition is a subclass of the **person** class. The **person** class is a structural class, and therefore not a valid superclass of the new alias definition.

Refer to *VSI Enterprise Directory Management* for details of customizing the schema, and defining new classes.

5.5.9. Too Many Structural Superclass Chains

A structural class can be derived from multiple superclass chains, but only one of those superclass chains can include structural classes. All but one of the superclass chains must be a chain of abstract classes.

This rule is enforced because a structural class is intended to characterize an object, and it would be illogical for an object to be characterized as, for example, both a person and a printer.

Amend your class definition so that only one of the superclasses listed is a structural class, all the rest being abstract classes. Then recompile the schema.

5.5.10. Wrong Kind of Object Class for Name Form

Name form s can only be defined for structural classes and alias classes. This error indicates that you have defined a name form for an auxiliary or abstract class. Remove the specified name form, and recompile.

5.5.11. Duplicate Keyword

You can customize the keywords that the DXIM command line interface accepts in commands to refer to attributes, classes, and syntaxes. These keywords must be unambiguous. If the schema compiler finds that a keyword is assigned to more than one definition, it displays an error.

Change the schema to remove duplicate keywords, and recompile.

5.5.12. Multiple Windows for Name Form

The schema has detected that more than one window definition has been specified for a given name form. Remove all but one of the window definitions, and recompile.

5.5.13. Cannot Open Input File

Check for the existence and file protection of the file that cannot be opened.

5.5.14. Cannot Write Schema Output File

Check the existence and protection of the directory into which the output file should be written, and check that there is sufficient disk space for the file.

5.5.15. Loop Detected While Processing

Some schema definitions, such as attribute sets, involve nested definitions. For example, an attribute set definition can list another attribute set. The schema compiler only allows ten levels of nested definition. If this limit is exceeded, the compilation fails and displays this error, indicating which definition contained too many levels of nesting.

Amend the relevant definition so that it uses fewer levels of nesting. For example, instead of referring to an attribute set, refer to the individual attributes within the set. When the nesting is no greater than ten levels deep, recompile the schema.

5.6. Cannot Create an Entry of a Specific Class (Motif Interface Only)

The DXIM windows interface relies on the directory schema for its window definitions. These windows definitions determine the information that is displayed in the DXIM windows, for example, what

attributes are displayed as part of the Create window. If the windows definition is missing in the directory schema file, DXIM cannot display an appropriate window.

If you have incorrect or missing windows definitions in your schema file, DXIM will be unable to display windows for the relevant classes. For example, the class name in the Create submenu will be displayed dimmed. Typical problems could be:

- Window definition is missing
- Window definition is present but necessary attributes are missing
- Window definition has inconsistent naming attribute(s)

Check that there is a window definition for each class of entry you are using. In addition, check that each class definition has the correct attributes defined, for example, all mandatory attributes for that class are defined. Refer to *VSI Enterprise Directory Management* for more information about windows definitions within the schema files.

5.7. You Want to Backup the Database While the DSA is Running

The files that comprise the DSA database are DXD$DIRECTORY:DSA-INFORMATION-TREE.*

When you backup the database, you are advised to backup all of these files. You might also backup the schema files from the same directory, for completeness. This means that if you restore the backup, you can also restore the schema, in case the schema has changed since the backup date.

If the DSA is not running, you can simply backup the most recent set of database files.

If the DSA is running, then the set of tasks you need to do depends on the system environment that the database files are in. See Section 5.7.1.

5.7.1. Backing Up on Systems without AdvFS Utilities

On OpenVMS systems, do the following to backup the database:

1. If you run the DSA with volatile modifications, turn the option off to flush all data to the disk, by entering the following command:

   ```
   ncl set dsa volatile modifications false
   ```

2. Find the last DIT check interval, then enter the following command:

   ```
   ncl show dsa dit check interval
   ```

   Note the DIT check interval so that you can reset it after the backup.

3. Set the interval to 0 to suspend DIT checking, as follows:

   ```
   ncl set dsa dit check interval 0
   ```

4. Then use the following commands to cause an immediate DIT check:

   ```
   ncl show dsa dit check last time
   ```

   Make a note of the last DIT check time.

```
ncl set dsa dit check unscheduled time
```

The unscheduled DIT check may take a few seconds to complete. To be confident that it has completed, use the **ncl show dsa dit check last time** command to see when its value changes from the one you made a note of.

When the DIT Check Last Time attribute shows that the unscheduled DIT check has completed, it is safe to backup the DSA database files.

5. Backup the database files.

6. When you have completed the backup, reset the DSA configuration.

   If you run the DSA with volatile modifications, turn the option back on by entering the following command:

   ```
   ncl set dsa volatile modifications true
   ```

7. Reset the DIT check interval, then, enter the following command:

   ```
   ncl set dsa dit check interval hh:mm
   ```

   where *hh:mm* is the interval that was in effect before you set it to 0 above, for example, 12:00. DIT checking resumes at the intervals you specify.

Chapter 6. Problems with Access Control and Security

This chapter describes those problems that relate directly to access control and security.

Some reasons for failing to access information are not access control or security problems, but problems with distributed operations (see Chapter 4) and DSA configuration. For example, if information seems to be inaccessible, check that the relevant DSAs are enabled, that the information actually exists, and so on, before assuming that it is a security problem. Access control and security problems are difficult to analyze and solve, so you should eliminate other possible causes first.

Table 6.1 lists the different types of security and access control problem that you might encounter.

Table 6.1. Problems with Security and Access Control

Problem	Refer to
User Cannot Access Directory Information As Expected	Section 6.1
Cannot Replicate Between DSAs	Section 6.2
User Receives Information that Is Known to Be Incomplete	Section 6.3
Authentication Is Not Successful	Section 6.4
Directory Returns an Unwilling to Perform Error	Section 6.5
Changing Security Configuration Seems to Have No Effect	Section 6.6
Need to Analyze Your Access Controls	Section 6.7
Need to Bypass Access Controls	Section 6.8

6.1. User Cannot Access Directory Information As Expected

If you have implemented the access controls suggested by VSI, then users should be able to read most directory information without authentication. Users should be required to authenticate themselves before being able to modify information.

Your first task is therefore to determine whether the user should have had access to the information that was denied to them. If the user was denied a particular type of access to particular information for good reason, then this is not a problem.

Assuming that you decide that the user has been denied access when they should have been granted access, you can use this section to try to diagnose the problem. Table 6–2 lists the most likely reasons why a user can fail to access information.

Table 6.2. Problems Accessing Information

Symptom	Refer to
User Has Insufficient Access Rights	Section 6.1.1
DSAs Do Not Trust Each Othe	Section 6.1.2

6.1.1. User Has Insufficient Access Rights

If you have implemented access controls, then a user might receive an error message when trying to access some types of information in certain ways.

Note that an error message is only returned if the entire request is denied to the user. If part of a user request is satisfied, then no error message is displayed. For example, if a user attempts to display all attributes, but succeeds in displaying only a few attributes, no error is displayed. The user might, nevertheless, complain that the Enterprise Directory has failed to give the access that they expect.

If a user receives the **Unwilling to Perform** error, then their access was prevented because the DSA that holds the relevant information has one or more characteristic attributes configured. See Section 6.5 for details of those attributes.

If a user receives the **Insufficient Access Rights** error message when trying to access information, or that they have been denied access to certain information, do the following:

1. Check whether the user authenticated themselves using their name and password.

   Some types of access to some types of information require authentication. If the user still cannot access the information when they have authenticated, then you need to look at the access controls for the naming context that contains the information they are trying to access.

   If you find that the relevant type of access to information requires authentication, and you think that this is too restrictive, then you might want to change access controls so that the same access is permitted without authentication. For example, it is reasonable to require authentication before permitting modifications of directory information, but perhaps too restrictive to require authentication before displaying some attributes, such as names and telephone numbers.

2. Check that the information actually exists.

   Bind to the Enterprise Directory, and authenticate yourself as a directory manager. As a directory manager you should be able to read all information, except for passwords. Therefore, you can check whether the information exists.

3. Check whether the user can access the information if they use a different DSA.

   Help the user to use the DXIM command line interface to bind to a different DSA and authenticate themselves. Then try to access the same information again. If the user was trying to modify a particular attribute, then ask them to attempt the same modification. If the user was trying to display information, ask them to try again using the **master information** control.

   If the master DSA allows access to the information, then it is possible that the user was receiving information from a shadow DSA, where the shadow DSA had different access controls.

   Also, ask the user to bind and authenticate to each shadow DSA for the information, and try to access the same information on each of them. If you get different access from different DSAs, then you need to revisit your implementation of access control so that such inconsistencies do not occur. Refer to *VSI Enterprise Directory Management* for details of how VSI recommends you to implement access controls. You should ensure that all shadow copies of each naming context are subject to the same access controls as the master copy of the naming context.

   If access controls are consistent in all relevant DSAs for the user, and allow the user to access the information when bound directly to a DSA that holds the information, then you probably have

a problem with your DSA security configuration. The DSA to which the user usually binds and authenticates might not be trusted by the DSA that holds the information they tried to access. The second DSA therefore does not assume that the user's authentication to the first DSA is reliable. See Section 6.1.2 for further details.

By now you should know that:

- The information exists
- Access control is consistent on all DSAs that hold the information
- The user is denied access even when they bind directly to each of the DSAs

If all this is true, then it is likely that the access controls for the information are denying access, rather than one of the preceding problems. Therefore, you need to look at the access controls for the naming context that contains the information the user attempted to read.

If you have kept a record of the access controls you implemented, then refer to those records, and analyze them as described in Section 6.1.1.2 to try to determine what is wrong with the access controls.

If you are not sure what access controls have been implemented, then you need to interrogate the Enterprise Directory to find that out, as described in Section 6.1.1.1.

6.1.1.1. Finding Out What Access Controls Are Implemented

This section describes how to find out what access controls apply to a naming context. Having done these tasks, you can then analyze the access controls to determine why a particular user request failed.

1. Find out which DSA is the master DSA for the information the user was trying to access.

   Refer to your configuration records, or use NCL commands to identify which DSA is master DSA for that information.

2. Use DXIM to bind to the master DSA for the relevant naming context, and authenticate yourself as a directory manager.

   If you have implemented different access controls for different naming contexts, then you need to authenticate yourself as a directory manager of the relevant naming context. If you cannot do this, then you need to refer this problem to someone who can.

3. Use the DXIM SEARCH comman d to locate all `prescriptiveACI` attributes in the relevant naming context.

   For example, if the user wanted to access information in the naming context called `/c=us/o=abacus/ou=sales`, use the following DXIM command[1]:

   ```
   dxim> search /c=us/o=abacus/ou=sales where prescriptiveACI=* subentries
   ```

   This command displays a list of all entries beneath the specified entry that contain the `prescriptiveACI` attribute.

   If the relevant naming context has subordinate naming contexts held on this DSA, then ignore any listed entries that are held within those subordinate naming contexts. You only need to analyze

[1]Note the use of the `subentries` control in the SEARCH command. For previous versions (before V3.0), omit the subentries control from the command.

`prescriptiveACI` attributes within the naming context that holds the information to which access was denied.

4. If the command returns no entries that are within the relevant naming context, then search your entire organization's DIT, for example:

   ```
   dxim> search /c=us/o=abacus where prescriptiveACI=* subentries no chaining
   ```

   This command restricts the search to naming contexts held locally. The DSA only uses local information when determining whether to permit access to its information.

5. If there are no entries containing the `prescriptiveACI` attribute at all, then access to the information should be almost unrestricted. The fact that the user was denied access must therefore be because the DSA has been configured using NCL commands so that it explicitly denies access to the user, or to all users, on a given node.

   The characteristic attributes of the DSA entity that relate to security are not the recommended way to implement security. VSI recommends that you use the `prescriptiveACI` attribute to control access, as described in *VSI Enterprise Directory Management*. VSI recommends that you delete any characteristic attributes that are causing access problems, unless they have been added for a specific reason. Refer to the NCL online help for the Directory Module for information about the characteristic attributes that affect security. Section 6.5 describes how these attributes can interfere with user requests that involve more than one DSA.

If your SEARCH commands find one or more entries that contain the **prescriptiveACI** attribute, then you need to analyze those attributes to see what access should be allowed for the user.

Use the DXIM SHOW command to display the value of each `prescriptiveACI` attribute within the relevant naming context. If there are no access controls in the naming context, display the values of each `prescriptiveACI` attribute in the immediate superior naming context. Again, if there are none, refer to the next superior naming context, and so on, until you find at least one `prescriptiveACI` attribute.

If you have followed VSI recommendations, then there is exactly one **prescriptiveACI** attribute in your DIT, and it is held in the naming context that contains the highest entries in your DIT. However, if access controls are causing problems or you have ignored the recommendation, you should search the DIT as described in the steps above to make sure that there are no other **prescriptiveACI** attributes that you were not aware of.

Having foun d the access controls that apply to a naming context, you can analyze them as described in Section 6.1.1.2.

6.1.1.2. Analyzing Access Controls

This section describes how to analyze the values of the `prescriptiveACI` attribute. This enables you to determine why a particular user request failed. See Section 6.1.1.1 for details of how to find the `prescriptiveACI` attributes in a given naming context.

When you are analyzing the `prescriptiveACI` attributes, you might find it useful to write the values to a file so that you can print them out. You might also find it useful to be able to use cut and paste to rearrange the order of the various access control information items (ACIitem s) into their order of significance.

When analyzing ACIitems always remember that access to attributes and values usually requires access to entries, and that some of the permissions have prerequisites. Refer to *VSI Enterprise Directory Management* or the DXIM command line interface online help for full details of access controls.

Also remember that the DSA always uses a distinguished name when determining access rights. If the user authenticates using an alias name, make sure that you use the relevant distinguished name during the problem solving steps described below.

Finally, remember that the basic rules of access control are to enforce the highest precedence, and the most specific reference to the user or to the information. ACIitems of lower precedence, or less specific reference to the user or the information, are overruled. For example, an ACIitem of high precedence that grants a specific user all access to a specific attribute value will overrule an ACIitem of low precedence that grants all users access to all information.

Analyze the ACIitems as follows:

1. Analyze the user request to find out what types of access permission it requires.

   For example, if the user has problems searching the directory, then the relevant permissions are Browse, Search, Read and Return Name. You can ignore the permissions that relate to modifying information.

   Similarly, if the user has problems renaming or modifying entries, then the relevant permissions are Rename, Remove Attributes and Values, and Add Attributes and Values. Other permissions can be ignored.

   If the user is having problems accessing particular attributes, then permissions that do not mention those attributes can be ignored.

   The more precisely you know what the user's problems are, in terms of the access permissions required, the easier it will be to determine which element of the ACIitems is causing the problem.

2. Arrange all ACIitems in order of precedence (and discard any that do not relate to the user's problem).

   Every ACIitem states a precedence. Put the highest precedence at the top of your list of ACIitems. ACIitems of equal precedence can be listed in any order.

3. Look at all of the ACIitems that share the highest precedence value to see whether any of them affect the user who complained of insufficient access rights.

   Typically, the ACIitem of the highest precedence is the ACIitem that defines access controls for directory managers.

   Try to identify the ACIitem or ACIitems that mention the user most specifically, as follows:

   i. If the user is not trying to access their own entry (the entry whose name they specified when authenticating), go to this step.

   ii. If the user was being denied access to their own entry, then look for the `OWNER` keyword.

      If you find the `OWNER` keyword, and the user was denied access to information in their own entry, look at the details of the ACIitem or ACIitems that have the `OWNER` keyword. If any of those ACIitems deny the relevant type of access to entries, or to the particular information that was concealed, then you have located the source of the problem. Note that if the user was trying to access their own entry, and an ACIitem with the `OWNER` keyword denies the relevant type of access to the information, then that denial overrules any grants specified in other ACIitems of the same or lower precedence.

If those ACIitems grant access to the information that was concealed, or do not mention the information at all, go on to the next step.

iii. Search for the distinguished name of the user in the `NAMES` arguments of each ACIitem.

If the user is mentioned explicitly in a `NAMES` argument, then look at the details of the relevant ACIitem or ACIitems. If any of those ACIitems deny the relevant type of access to entries, or to the particular information that was concealed, then you have located the source of the problem. Note that any denial stated in an ACIitem that explicitly names the user overrules any grant specified in any other ACIitem of the same precedence.

If those ACIitems do not mention the information at all, go on to the next step.

iv. Search for any ACIitems that have the `GROUP` keyword.

If any ACIitems have the `GROUP` keyword, make a note of all listed group names. Use DXIM to see whether those `groupOfNames` entries are held by the DSA that failed to provide the desired access. For example, if an ACIitem mentions a group called `/c=us/o=abacus/cn="Management Team"`, bind to the DSA and use the following DXIM command:

```
dxim> show /c=us/o=abacus/cn="Management Team" no chaining
```

If the DSA has a copy of the entry, it displays it and shows the names of all members of the group. If the DSA does not have a copy of the entry, then it does not attempt to contact other DSAs to access the entry. The user is assumed not to be a member of the group, and any permissions granted to members of that group do not apply to the user when bound to this DSA.

If any ACIitem mentions a group that is held by the DSA and the user is listed as a member of the group, then that ACIitem applies to the user.

If you find such an ACIitem, look at the details. If any such ACIitems deny access to entries, or to the particular information that was concealed, then you have located the source of the problem.

If the user is not a member of a group specified in any ACIitem held by the DSA, or such ACIitems do not mention the information or type of access that the user was denied, then go to the next step.

v. Search for any ACIitems that have the **BEGINNING** keyword.

If any ACIitems have the `BEGINNING` keyword, look at all the distinguished names listed with that keyword.

If any of the distinguished names are names of entries directly superior to the user's distinguished name, then the ACIitem might apply to the user. For example, if the user has the name `/c=us/o=abacus/ou=sales/cn="tom thomas"`, then they would be affected by an ACIitem that has `BEGINNING /c=us/o=abacus/ou=sales`.

If the `BEGINNING` clause for the relevant name also has the `MINIMUM` and/or `MAXIMUM` keywords, then the user is only affected if his name falls within the limits specified by those keywords. For example, Tom Thomas' entry is one level beneath the entry listed in the `BEGINNING` clause. Therefore, if the clause also states `MINIMUM 2`, he is not affected by the ACIitem, because the ACIitem only applies to entries that are two or more levels beneath the listed entry.

If the `BEGINNING` clause for the relevant name specifies no minimum or maximum, then all entries beneath the specified entry, and the specified entry itself, are affected by the ACIitem.

If you find any ACIitems that affect the user, look at the details to see whether they deny the relevant type of access to the information. If so, you have foun d the source of the problem.

If no such ACIitems are specified, or none of them apply to the user, or none of them deny the relevant type of access, then go to the next step.

vi. If none of the above steps help you to locate the source of the problem, then all that remains is to search for an ACIitem that contain s the `USERS ALL` keywords.

If there are any such ACIitems, look at their details to see whether any of them deny the relevant type of access to information. If so, you have foun d the source of the problem.

vii. If none of the ACIitems of this precedence help you to determine the source of the problem, then you need to refer to all ACIitems of the next highest precedence.

Return to the beginning of this list of steps, and repeat the analysis for all ACIitems of the next highest precedence.

If, after analyzing all ACIitems, you still have not foun d an ACIitem that applies to the user, and grants them the relevant type of access to the information, then that is the source of the problem. In the absence of any relevant grant, the Enterprise Directory denies access.

If you find an ACIitem that grants access to the user, but access is still being denied, then there must be an ACIitem that denies access at equal precedence. You need to find the ACIitem that is explicitly denying access.

If you cannot find such a denial, then it is possible that some of the characteristic attributes of the DSA entity are affecting the user's access. In this case, the user should have received an `Unwilling to Perform` error, as mentioned in the introduction to this section. See Section 6.5.1 for further details.

Having analyzed the ACIitems, and determined why the user is being denied access to information, you need to replan your ACIitems to implement the required controls.

Because analyzing ACIitems is so complicated, VSI strongly recommends that you have as few ACIitems as possible, and that you store them all as values of a single `prescriptiveACI` attribute. This at least makes the task of analyzing them easier, reduces the chances of overlooking any ACIitems, and reduces the potential for ACIitems to contra dict each other.

6.1.2. DSAs Do Not Trust Each Other

If a user requests a service that requires communication between DSAs, then this can interfere with the user's level of authentication.

When a user authenticates to the Enterprise Directory, by supplying a name and password, they are actually authenticating to the particular DSA to which they are bound. If that DSA needs to get information from other DSAs, the other DSAs do not necessarily assume that the authentication claimed by the first DSA is reliable.

Whenever two DSAs communicate they determine whether they trust each other. If they trust each other, then they both assume that any user who has authenticated to one DSA can be considered to

have authenticated to both. If two DSAs do not trust each other, then they do not assume that each other's users are adequately authenticated. Thus, if a user attempts to access information that requires communication between DSAs, they might find that they can access less information than they expected.

VSI recommends that you configure all DSAs to trust each other, as described in *VSI Enterprise Directory Management*. If you do not configure this trust, then you severely limit the ability of your users to have consistent access to directory information. Users will get different responses depending on which particular DSA they are bound to.

To determine whether two DSAs trust each other, do the following:

1. Use NCL to display the AE titles of the two DSAs.
2. Use the DXIM command line interface to bind to one of the two DSAs authenticated as a directory manager.
3. Use the DXIM SHOW command to try to display each of the DSA entries, where the distinguished name of each entry is the same as the DSA's AE title. Specify that you want to see the `trustedDSAname` attribute. For example:

   ```
   dxim> show /c=us/o=abacus/cn=DSA1 attribute trustedDSAname no chaining
   dxim> show /c=us/o=abacus/cn=DSA2 attribute trustedDSAname no chaining
   ```

   Note the use of the `no chaining` control, because a DSA only uses local information when trying to determine whether it trusts another DSA.

   If both DSA entries have the `trustedDSAname` attribute, check that the values of the two attributes state the distinguished name of the relevant DSA entry itself. For example, the display for the two DSA entries should be as follows:

   ```
   dxim> show /c=us/o=abacus/cn=DSA1 attribute trustedDSAname no chaining

   /C=us/O=abacus/CN=DSA1
       Trusted DSA Name = /C=us/O=abacus/CN=DSA1

   dxim> show /c=us/o=abacus/cn=DSA2 attribute trustedDSAname no chaining

   /C=us/O=abacus/CN=DSA2
       Trusted DSA Name = /C=us/O=abacus/CN=DSA2
   ```

4. If either of the commands fails, or shows no `trustedDSAname` attribute, or has some other value of that attribute, then that is the reason why the two DSAs do not trust each other.

   Refer to *VSI Enterprise Directory Management* for details of how to create entries to represent DSAs.
5. Use DXIM to bind to the other DSA, and repeat the above steps to see whether that DSA has the two DSA entries with the **trustedDSAname** attributes correctly specified.
6. If those entries exist on both DSAs, and appear to be correctly defined, then it is possible that the values in the `userPassword` attributes of the two DSA entries do not match the Password attributes of the two DSAs.

   Use NCL and DXIM to reset the passwords of both DSAs manually. Set the Password attribute of the DSA entity and the `userPassword` attribute of the entry that represents the DSA in the DIT. Set the same value in both places.

   Make sure that the naming context that contains the DSA entry is updated on all shadow DSAs as soon as possible so that all DSAs are aware of the new passwords of the two DSAs.

6.2. Cannot Replicate Between DSAs

This section only describes those replication problems that relate directly to security. For general replication problems, refer to Section 4.5. If replication fails due to a security problem, this could be caused by one of the problems listed in Table 6.3.

Table 6.3. Problems with Replication

Symptom	Refer to
Supplier DSA Cannot Verify the Identity of the Consumer DSA	Section 6.2.1

6.2.1. Supplier DSA Cannot Verify the Identity of the Consumer DSA

A consumer DSA specifies its AE title and password when it contacts the supplier DSA. The values specified by the consumer DSA are the AE Title and Password characteristic attributes of its DSA entity.

The supplier DSA attempts to compare the specified password with the `userPassword` of the entry that has the same name as the specified AE title.

If the supplier DSA finds no such entry, or finds that the passwords do not match, then it does not supply any directory information to the consumer DSA, and the consumer DSA displays the following error:

```
Supplier DSA is unavailable.
```

You need to find out exactly why the supplier DSA refused to supply a copy of its naming contexts. To analyze and solve this problem, do the following:

1. Check the AE Title of the consumer DSA, as follows:

   ```
   ncl> SHOW DSA AE TITLE
   ```

2. Use the DXIM command line interface to try to verify that there is an entry representing the consumer DSA, that the entry holds the correct password, and that the entry is held on the supplier DSA, as follows:

   a. Bind to the supplier DSA, authenticate yourself as a directory manager, and try to show the entry. For example, if the consumer DSA's AE title is `/c=us/o=abacus/cn=DSA1`, then use the following DXIM command:

      ```
      dxim> show /c=us/o=abacus/cn=DSA1 no chaining
      ```

      If this command displays the entry, then the entry is held on the supplier DSA. This probably means that the password in the entry does not match the Password attribute of the consumer DSA. You need to take steps to ensure that the passwords match. If you know the DSA's password, then you can use the DXIM COMPARE comman d to see whether the value in the directory is what you think it should be. If the passwords do not match, then replication is not the only DSA activity that will fail.

   b. If the preceding command does not display the entry, remove the `no chaining` control, and repeat the command:

      ```
      dxim> show /c=us/o=abacus/cn=DSA1
      ```

If this command displays the entry, then the entry exists, but is held on a remote DSA. In order for replication to succeed, the supplier DSA must have a local copy of the entry representing the consumer DSA. See *VSI Enterprise Directory Management* for advice about how to implement DSA trust.

c. If neither of the preceding DXIM SHOW commands displays the entry, then use DXIM to bind directly to the DSA that is supposed to be the master DSA for the entry, and use another DXIM SHOW command to verify whether the entry exists. Authenticate yourself as a directory manager so that access controls do not prevent you from showing the entry.

   If the command does not display the entry, then the entry does not exist. You therefore need to create the entry, and arrange for it to be replicated to the supplier DSA (assuming the supplier DSA is not the master DSA for the entry). Refer to *VSI Enterprise Directory Management* for details of how to plan and create DSA entries.

   If the command displays the entry, then you have discovered a second problem. Firstly, the entry needs to be replicated to the supplier DSA, so that it can verify the identity of the consumer DSA. Secondly, one or more of your DSAs has incomplete or inaccurate knowledge information, causing the SHOW command to fail. To solve the knowledge information problem, see Chapter 4.

If your analysis of the problem is that the password specified by the consumer DSA does not match the `userPassword` of the DSA entry in the DIT, then take the following steps:

1. Use the NCL SET DSA command to set the Password attribute of the consumer DSA, as follows:

   ```
   ncl> SET DSA PASSWORD "password"
   ```

2. Use DXIM to set the `userPassword` attribute of the DSA's entry so that it matches the one specified in NCL.

   ```
   dxim> set password /c=us/o=abacus/cn=DSA1
   Old_Password>
   New_Password>
   Verify_Password>
   ```

   You need to authenticate as a directory manager to use the DXIM SET PASSWORD command successfully. You also need to know the current value of the `userPassword` attribute. If there is no `userPassword` attribute, you can simply type RETURN at the `Old_Password>` prompt.

   If you do not know the current value of the `userPassword` attribute, you need to:

   a. Set the DXIM password checking control off.

   b. Rebind to the Enterprise Directory.

   c. Use the DXIM MODIFY command to remove the existing `userPassword` attribute.

   d. Add a new **userPassword** that matches the one specified in the preceding NCL command. For example:

      ```
      dxim> set default no password check
      dxim> bind link2 name /c=us/o=abacus/cn="John Manager" password
      Password>
      dxim>
      dxim> modify /c=us/o=abacus/cn=DSA1 -
      ```

```
_dxim> remove attribute userPassword -
_dxim> add attribute userPassword="password"
dxim>
dxim> unbind link2
dxim> set default password check
```

3. Update all shadow DSAs for the naming context that contains the entry you modified.

6.3. User Receives Information that Is Known to Be Incomplete

This could be caused by any of the problems in Table 6.4.

Table 6.4. Incomplete Information

Symptom	Refer to
Access Controls Are Denying Access to Some Information	Section 6.3.1

6.3.1. Access Controls Are Denying Access to Some Information

If a user complains that some of the information they requested was not returned from the directory, then there are two likely causes.

Firstly, the request might have been satisfied by a DSA that has an incomplete copy of the information. This is not a security problem; see Chapter 5.

The second reason is that access controls are preventing the DSA from returning all of the information requested. See Section 6.1 for a full description of how to diagnose and solve this problem.

6.4. Authentication Is Not Successful

This could be caused by any of the problems listed in Table 6.5.

Table 6.5. Problems with Authentication

Symptom	Refer to
Username Missing or Incorrect	Section 6.4.1
Password Missing or Incorrect	Section 6.4.2
DSA Cannot Find the User's Entry	Section 6.4.3

6.4.1. Username Missing or Incorrect

You can define a default username to be used by DXIM on Bind requests to the DSA. This is defined in the DUA defaults file in your home directory, in the parameter `DUA.Requestor`.

The name defined in the **DUA.Requestor** parameter acts as the default distinguished name for the following:

- The Name parameter in a DXIM command line interface Bind command
- The User's Distinguished Name field within the Authenticate... window of the DXIM windows interface

A user can also specify a name explicitly if they do not want to use the default.

In any case, if the username provided during binding and authentication is not a valid name of an existing directory entry, then the bind and authentication fails. This is indicated by an Authentication Failure event being issued with the Unknown User reason (see Section 10.1.16).

The bind and authentication also fails if the DSA to which you are binding cannot access the entry that you specify to compare passwords.

If a user reports that they are having problems authenticating, check that they are using a valid distinguished name, and check that the distinguished name actually names an entry.

If the user is using the default name, and the name is invalid, edit the DUA defaults file to specify a valid distinguished name.

6.4.2. Password Missing or Incorrect

When a user authenticates to the directory, they can specify a password. If the DSA to which they are binding can verify that the password supplied matches the `userPassword` attribute of the user's directory entry, then the DSA gives the user greater access to directory information.

The user can specify a password when authenticating, either by typing the password into the password field of the Authenticate... window, or by using the `PASSWORD` argument in a DXIM BIND command.

If the password specified by the user does not match the password in the user's entry, then the authentication fails.

If a user reports that their password is being rejected by the Enterprise Directory, use the DXIM COMPARE command to check the password against the user's directory entry. For example, if a user has a directory entry called `/c=us/o=abacus/ou=sales/cn"Jon Low"`, and claims to have the password "mumble", use the following DXIM command:

```
dxim> compare /c=us/o=abacus/ou=sales/cn"Jon Low" with userpassword=mumble
```

Remember that the password attribute is case sensitive, so make sure that you use the same combination of upper case and lower case letters as the user.

The COMPARE comman d confirm s whether the password is correct. If not, use the DXIM SET PASSWORD command to change the user's password. They should then be able to use their new password to authenticate.

If the password is correct, then it is possible that the DSA cannot access the user's entry in order to do the password comparison. To verify this, bind to the DSA, and attempt to show the user's entry. If the command fails, this indicates that at least one of your DSAs has incomplete knowledge information. See Chapter 4 for details of solving knowledge information problems.

If the command succeeds, it is possible that the access controls specified for the `userPassword` attribute are preventing the DSA from doing the password comparison. To verify this, see Section 6.1 for details of solving access control problems.

6.4.3. DSA Cannot Find the User's Entry

If both the name and the password supplied by the user are correct, then the authentication might fail because the DSA cannot find the user's entry. This indicates one of the following problems:

- The DSA or DSAs that hold the user's entry are unavailable.

  In this case, the user must wait for one or more of the relevant DSAs to become available so that their identity can be verified.

- The DSA is prohibited from chaining, and the user's entry is not held on the DSA.

  In this case, the prohibition of chaining prevents the DSA from verifying the user's identity. Check the configuration of the DSA to see whether the DSA is prohibited from chaining, as follows:

  ```
  ncl> SHOW DSA PROHIBIT CHAINING
  ```

  If the value of this attribute is TRUE, then the DSA will always fail to authenticate user's whose entries are not held on the DSA. If that is not acceptable, set the value of this attribute to FALSE, or make sure that this DSA has copies of all user entries.

- One or more of your DSAs has incomplete or incorrect knowledge information.

  To verify this, bind to the DSA without authentication, and use the DXIM SHOW command to try to display the user's entry. If the command fails, then you have a knowledge information configuration problem. See Chapter 4.

- Your DSA has insufficient access privileges to access the user's entry to verify the password.

  If the user's entry is held on another DSA, then it is possible that the access controls implemented on that other DSA deny the user's DSA compare access to the **userPassword** attribute. This is an access control problem. See Section 6.1 for details of how to diagnose and solve access control problems. It is important that your DSAs have sufficient access rights to be able to check users' passwords.

6.5. Directory Returns an Unwilling to Perform Error

This could be caused by any of the problems listed in Table 6.6.

Table 6.6. Directory Unwilling to Perform

Symptom	Refer to
DSA Entity Configuration Is Preventing Access	Section 6.5.1

6.5.1. DSA Entity Configuration Is Preventing Access

The only reason a HP DSA ever returns the `Unwilling to Perform` error message (reported by DXIM as `The Directory is Unable to Perform the Command`) is because that DSA has one or more of the following DSA characteristic attributes configured using NCL:

Writer Names
Reader Names
Writer NSAPs

Reader NSAPs
Trusted DSA Names
Read Only DSA Names
Trusted DSA NSAPs
Read Only DSA NSAPs

To solve the problem, you must first identify which DSA returned the error, as follows:

1. If the user tried to access information on their default DSA, that is, the DSA to which they are bound, then that is the DSA that returned the error.

   You can determine which DSA a user is bound to by reference to the `DUA.KnownDSAs.ae_title` and `DUA.KnownDSAs.paddr` parameters in their DUA defaults file. If the user does not have a DUA defaults file in their home directory, refer to the system DUA defaults file.

   You can determine whether the information was held on the default DSA, by using the DXIM command line interface to recreate the user request, and specifying the `no chaining` control. For example, to recreate a typical SHOW command, you might use the following command:

   ```
   dxim> show /c=us/o=abacus/ou=research/cn="Clive Barnard" no chaining
   ```

   If this command returns the `Unwilling to Perform` message, then it is the default DSA that is causing the problem. Use the following NCL command to display the characteristic attributes of the default DSA:

   ```
   ncl> SHOW DSA ALL CHARACTERISTICS
   ```

   Look at the attributes listed at the beginning of this section to see whether the user is mentioned, or whether the NSAP used by the user is mentioned. If such attributes exist, but the user is not mentioned, then this is the cause of the problem. You need to reconfigure the attributes to give the user the access rights they require.

   Note that if you reconfigure these attributes, the new values only apply to new connections to the DSA. Existing connections are unaffected. You need to disable and re-enable the DSA to ensure that all existing connections are terminated, so that the new security configuration applies to all connections.

2. If the DXIM command in the previous step returned a continuation reference, then the user request involved chaining. For example:

   ```
   dxim> show /c=us/o=abacus/ou=research/cn="Clive Barnard" no chaining

   Continuation references
     Target Object:    /C=us/O=Abacus
     Operation phase:  Proceeding
     Next RDN:         2
     Reference type:   Subordinate reference
       DSA Name:       /C=us/O=Abacus/CN=DSA6
       DSA Address:    "DSA"/"DSA"/"DSA"/NS+49AA0019922AA2200000,CLNS
   ```

   This suggests that the user's default DSA was willing to perform the operation, but the DSA mentioned in the continuation reference was not.

   In this example, the message may be returned because /CN=DSA6 has the characteristic attributes listed above, but the attributes do not mention the user's default DSA. It is the user's default DSA that is being denied access to /CN=DSA6 in this case, rather than the user.

You need to look at the characteristic attributes of the remote DSA, and reconfigure them to list the user's default DSA as a trusted DSA.

3. If the configuration of the DSA mentioned in the continuation reference does not explain the problem, and it appears that the DSA should have been willing to process the user request, then you need to follow the user request further. The DSA listed in the continuation reference must have chained to a third DSA, and that third DSA might have returned the `Unwilling to Perform` message.

   To follow the user request, bind directly to the DSA listed in the continuation reference, and make the same user request, using the `no chaining` control. If the command returns a continuation reference, look at the characteristic attributes of the DSA listed in it, to see if they explain the problem.

4. Repeat this process until you identify the DSA that returned the message.

5. Reconfigure that DSA so that it supports distributed operations properly.

   This involves amending the value of the relevant characteristic attribute, to include the name or NSAP of the DSA that was not given access. A better solution is to implement the recommendation that all DSAs are represented by `decDSA` entries that are replicated to all DSAs, as described in *VSI Enterprise Directory Management*.

   If DSAs do not interwork freely, then user requests are not the only operations that will fail.

VSI suggests that you remove all eight of these attributes completely from all of your DSAs. They are not the recommended way of controlling access to directory information, and overrule the recommended access control functionality. They are also difficult to troubleshoot. See *VSI Enterprise Directory Management* for details of the recommended way to implement access controls, and for a discussion of the disadvantages of using these characteristic attributes.

This problem highlights the difficulties of tracing a user request across multiple DSAs. If you have distributed and replicated your DIT as recommended by VSI, then most user requests can be satisfied with minimal DSA interworking. You should find that the DSA that caused the problem was contacted directly by the user's default DSA, rather than being the last DSA in a complicated chain of communication.

6.6. Changing Security Configuration Seems to Have No Effect

If you use characteristic attributes of the DSA entity to reconfigure security, you might find that the configuration seems to have no effect. This is because the DSA only refers to those attributes when it receives a new connection. Once a connection has been accepted, the DSA never reassesses the security level of the calling application. Any changes to these attributes has no effect on existing connections.

To make sure that all connections conform to the new security policy, you need to disable and re-enable the DSA. This terminates all existing connections, and requires applications to reconnect to the DSA. The new connections will be checked against the new security configuration.

Note that this behaviour does not apply to changes to prescriptive access controls, which are the recommended way to implement security. Prescriptive access controls are checked for each request, and therefore take effect as soon as you change them, even for existing connections. Users can therefore find

that their access rights to some information has changed from one request to the next, even using an uninterrupted connection.

6.7. Need to Analyze Your Access Controls

For certain access problems you will need to analyze access controls to try to determine exactly why a user is being denied the right to do something that they should be allowed to do, or conversely, granted the right to do something they should not be allowed to do.

See Section 6.1 for details of how to analyze access controls.

6.8. Need to Bypass Access Controls

In certain circumstances you might need to bypass access controls. For example, if you make a mistake when setting up access controls, you might find that you have no access to the `prescriptiveACI` attribute, and are therefore unable to correct your mistake. Furthermore, you might find that no one has access to any directory information.

If you need to bypass access controls as an emergency measure, do the following:

1. Use NCL on the DSA that you want to access, to specify a Trusted DSA Name.

   For example, use the following NCL command:

   ```
   NCL> ADD DSA TRUSTED DSA NAME = {"/c=us/o=abacus/cn=Emergency Access"}
   ```

2. Create an Accessor entity as a subentity of the DSA to which you want to bind.

   Specify the same name as configured in step 1, and a password for that name. For example:

   ```
   ncl> CREATE DSA ACCESSOR "/c=us/o=abacus/cn=Emergency Access" -
   _ncl> PASSWORD "secretpassword"
   ```

   You can now use DXIM to bind and authenticate to the DSA using the name and password. It is not necessary for the specified name to actually exist in the DIT. The DSA considers you to be a trusted DSA for the duration of the binding, and allows you to modify the `prescriptiveACI` attribute. Delete the Accessor entity and remove the Trusted DSA Name after modifying the `prescriptiveACI` attribute.

   Note that the Accessor entity is volatile data. If you delete the DSA entity, then the Accessor entity is lost permanently. It is not advisable to keep Accessor entities longer than necessary.

Chapter 7. Problems With Resources

This chapter covers problems associated with the system resources needed to operate the Enterprise Directory.

Resource problems are indicated by the Resource Exhausted event being issued and the Exhausted Resource counter being incremented. In addition, some DXIM and NCL error messages indicate resource shortages.

Table 7.1 lists possible resource problems.

Table 7.1. Problems with System Resources

Problem	Refer to
DSA Process Quotas	Section 7.1
DSA Cannot Load DIB fragment	Section 7.2
Replication Fails with No Resources Available	Section 7.3
Create or Enable DSA Fails with No Resources Available	Section 7.4
DXIM Fails with Insufficient Memory Error	Section 7.5
DIB Fragment Becomes Excessively Large	Section 7.6

7.1. DSA Process Quotas

Table 7.2 shows the recommended minimum settings for the DSA process quotas. For each quota, the table shows the following:

- The name of the quota as displayed by a `DCL SHOW PROCESS /QUOTA` command.
- The UAF name of the quota as used in the Authorize utility (for reference).
- The RUN qualifier used in the DSA startup file to set the quota.
- The value set by default in the DSA startup file.

Table 7.2. DSA Process Quotas

AST	**ASTLM**	**AST_LIMIT**	**100**
Buffered I/O byte count	BYTLM	BUFFER_LIMIT	variable[1]
Buffered I/O limit	BIOLM	IO_BUFFERED	100
Direct I/O limit	DIOLM	IO_DIRECT	18
Timer queue entry	TQELM	QUEUE_LIMIT	100
Open file	FILLM	FILE_LIMIT	100
Enqueue	ENQLM	ENQUEUE_LIMIT	150
Working Set Extent	WSEXTENT	EXTENT	variable[b]
Paging file	PGFLQUOTA	PAGE_FILE	variable[c]

[1]Set to 10% of the value of the NPAGEVIR SYSGEN parameter, or to 100000, whichever is higher.

[b]Set to the value of the WSMAX SYSGEN parameter.

[c]Set to the value of the VIRTUALPAGECNT SYSGEN parameter.

Check the resources available to the DSA as follows:

1. Use **SHOW SYSTEM** to find out the process identifier of the DXD$DSA-S ERVER process.

2. Use **SHOW PROCESS/QUOTA/ID=*pid*** where ***pid*** is the process identifier obtained in step 1.

The display shows what resources are still available to the process. If any resource is very low compared to the process defaults, modify SYS$STARTUP:DXD$DSA_STARTUP.COM to increase the defaults. If the page file quota is low, use AUTOGEN to reconfigure the system to support the DSA better.

After amending any resource, stop and restart the DSA to take advantage of the new settings.

7.2. DSA Cannot Load DIB fragment

When a CREATE DSA directive is issued, the DSA reads its DIB fragment from disk into memory. If this operation fails, it may be due to any of the reasons shown in Table 7.3.

Table 7.3. Problems Loading the DIB Fragment

Symptom	Refer to
The DSA Information Tree is Corrupt	Section 2.4.2
The DSA Information Tree is Incompatible with this Version of the DSA	Section 2.4.3
No Resource Available	Section 7.2.1

7.2.1. No Resource Available

If there is insufficient memory available for the DSA to load its DIB fragment, the NCL **No Resource Available** error message is issued. The error can also mean that the transport templates required by the Enterprise Directory do not exist.

If the DSA is installed on an OpenVMS system, or the problem occurs even after the templates have been created, read the rest of this section.

Check the memory available by entering the following command:

```
/sbin/swapon -s
```

This shows you how much swap space is available and how much is currently being used. The DSA will start to use the swap space as soon as it runs out of physical memory. If necessary, allocate more memory to the DSA.

Check the Buffered I/O byte count and Paging file quotas available to the DSA. Increase the quotas if necessary. See Section 7.1 for details.

Check that there is not another application using the DSA's LDAP port.

7.3. Replication Fails with No Resources Available

This could be due to any of the reasons shown in Table 7.4.

Table 7.4. Problems with Replication

Symptom	Refer to
Insufficient Disk Space	Section 7.3.1
Insufficient Memory	Section 7.3.2

7.3.1. Insufficient Disk Space

The shadowed DIB fragment is stored on disk at both the supplier and consumer DSAs. Check that there is sufficient disk space available at both systems. The disk requirements for the supplier and consumer are different.

The supplier DSA creates a file in the SYS$SCRATCH area. The DSA writes all information for all naming contexts that are to be supplied into that file, and maintains the file until replication is complete. It is therefore the disk space of SYS$SCRATCH that you need to check and increase if necessary. Also, check that those directories exist and are accessible to the DSA.

The consumer DSA also uses the SYS$SCRATCH area, but the file only ever contains details of a single naming context. If the consumer is to receive many naming contexts, each one is received and written to the temporary file individually.

You can edit the command line in DXD$DSA_STARTUP_INPUT.COM that defines the SYS $SCRATCH logical for the DSA process.

7.3.2. Insufficient Memory

Replication fails if there is insufficient memory available to carry out the operation on either the supplier DSA or the consumer DSA.

Check the Buffered I/O byte count and Paging file quotas available to the DSA. Increase the quotas if necessary. See Section 7.1 for details.

If you have insufficient memory available on the consumer DSA system, consider reconfiguring the supplier so that fewer or smaller naming contexts are supplied.

7.4. Create or Enable DSA Fails with No Resources Available

This problem can be caused by any of the reasons shown in Table 7.5.

Table 7.5. Problems Creating or Enabling a DSA

Symptom	Refer to
Insufficient Memory	Section 7.4.1
OSI Transport Entity Not Available	Section 7.4.2
Insufficient Process Quotas	Section 7.4.3
Local Access Point Establishment Failure	Section 7.4.4
Internal Software Error	Section 7.4.5

7.4.1. Insufficient Memory

Check that there is sufficient memory to create or enable the DSA.

Check the Buffered I/O byte count and Paging file quotas available to the DSA. Increase the quotas if necessary. See Section 7.1 for details.

7.4.2. OSI Transport Entity Not Available

Check that the OSI Transport entity is enabled and ready for communication as follows.

Run NCL and enter the following command:

```
NCL> SHOW OSI TRANSPORT STATE
```

The OSI Transport entity must be in state ON for the DSA to be created or enabled, unless the DSA is configured to use RFC1006 connections only.

7.4.3. Insufficient Process Quotas

There are insufficient process quotas to create or enable the DSA. See Section 7.1 for information about the quotas required by the DSA process.

7.4.4. Local Access Point Establishment Failure

The DSA cannot use its configured presentation address. Check for the Local Access Point Establishment Failure event.

This is returned under two circumstances:

- The DSA's presentation address is wrongly specified.

  Check the presentation address used by the DSA to ensure it is correct.

- There is a problem with the network.

  Refer to Chapter 3.

7.4.5. Internal Software Error

There is an internal software error. Contact VSI.

7.5. DXIM Fails with Insufficient Memory Error

Check that there is sufficient memory for DXIM to operate.

The user has insufficient memory available to run DXIM. Increase the memory available to the user by increasing the Paging file quota of the user's process.

7.6. DIB Fragment Becomes Excessively Large

If your DIB fragment becomes excessively large, such that you are running out of system resources, there are a number of options you can take as shown in Table 7–6.

Table 7.6. Problems with the Size of a DIB Fragment

Symptom	Refer to
Increase Disk Space	Section 7.6.1
Reduce Shadowing	Section 7.6.2
Reduce the Use of Indexes in the DSA	Section 7.6.3

7.6.1. Increase Disk Space

Consider moving to a different device.

Follow the instructions outlined in the DXD$LOGICALS_STARTUP.COM file, which is located in the SYS$STARTUP directory.

7.6.2. Reduce Shadowing

If your DSA is a consumer of information for too many supplier DSAs, or the information that it is consuming suddenly increases in size, you may run into resource problems.

Reduce the number of shadow naming contexts held by the DSA.

7.6.3. Reduce the Use of Indexes in the DSA

The DSA can maintain indexes of attribute values to improve performance. However, an index requires additional memory and disk space. A typical indexed attribute requires approximately 20 bytes of memory more than if it is not indexed.

If a given DSA has resource problems, and you cannot solve them using either Section 7.6.1 or Section 7.6.2, then you might consider reducing the DSA's use of indexes.

The DSA creates indexes in accordance with two lists in the schema. The two lists contain the names of attribute types for which the DSA should maintain an index. By removing attributes from those lists, you can save some memory and disk space. However, reducing the DSA's use of indexes will reduce its performance, especially for search requests.

See *VSI Enterprise Directory Management* for details of the schema files, and how to control indexing. Note that the decision to use indexes can vary from one DSA to another. If you have a particular DSA that has resource problems, you can consider reducing its use of indexes without impacting other DSAs that hold the copies of the same directory entries.

If you change the schema for this reason, you need to recompile the schema and then delete and recreate the DSA. During recreation, the DSA will discard the unwanted indexes.

Chapter 8. The DSA Accounting Facility

This chapter describes the DSA accounting facility. It contains two types of information. Section 8.1 contains information relevant to the person responsible for managing and configuring the accounting facility. Section 8.2 contains information relevant to the person responsible for processing the accounting file.

To fully understand the information in this chapter you need to be familiar with ASN.1 Basic Encoding Rules (ASN.1 BER) and with the ITU-T X.500 Recommendations.

If you enable the accounting facility, the DSA logs information about every subsequent user request it processes. This information is stored in binary format in an accounting file created by the accounting facility. The DSA does not decode or process the accounting information in the accounting file. If you decide to use the accounting facility, you need to provide a utility that can decode the binary format accounting file.

The accounting facility records detailed information about each user request. This information includes, for example, the time taken to process the request, and the type of request. If required, you can configure the accounting facility to store the full protocol data unit of user requests and errors.

It is likely that you will not need to use all the information the DSA records. You therefore need to decide what information you are going to use for your accounting purposes. For example, you may decide to bill people according to the amount of time it takes the DSA to process their requests. Your decision may mean that you need to set the Accounting Options characteristic attribute to include information about a user request or returned error. When you have decided what information to use as the basis of your accounting, you need to provide a utility that can extract the required information from the accounting files produced by the DSA.

The accounting facility is an optional service that is enabled by use of the Accounting State status attribute of the DSA entity (see the Directory Module online help for more information). The accounting facility is included in the DSA for use by those who have a requirement for accounting information. If you enable accounting, the DSA performance is impacted and a large amount of disk space may be consumed by accounting files. Each record written to the accounting file consumes, on average, 162 bytes of disk space. This amount is increased if you use the Accounting Options characteristic attribute to include either the request or error protocol data units in accounting records. In order to reduce the impact of enabling accounting, VSI recommends that you store accounting files on a different disk to the one that contains the DSA files (see Section 8.1.3 for more information).

When the accounting facility is enabled, each time the DSA processes a user request, for example, a search, it writes a record of the operation to the accounting file. Each record is written in binary. The format of the accounting records is defined in the ASN.1 Basic Encoding Rules (BER) format. ASN.1 is a standard form of notation used to describe the format of a data file. There are six different types of record that the DSA can write to the accounting file. The different types are described in Section 8.2.1.

You can use characteristic attributes of the DSA entity to control how often the accounting facility closes the current accounting file and creates a new one. This is called accounting file rollover. When the accounting file is closed, you can use your decoding utility to process the file as required. You cannot decode or process the current accounting file, that is, the accounting file to which the DSA is currently writing accounting information. See Section 8.1.4 for more information about managing accounting file rollover.

The tasks involved in setting up and maintaining the accounting facility can be divided into two types; managing the accounting facility, and processing the accounting file. The person responsible for managing the accounting facility needs to have an understanding of NCL, including knowledge of NCL events and attributes. The person responsible for processing the accounting file needs to have a knowledge of ASN.1 encodings.

8.1. Managing the Accounting Facility

This section contains information needed by the person responsible for managing the accounting facility, and describes the tasks involved.

8.1.1. Enabling and Disabling the Accounting Facility

To enable accounting, enter the following command:

```
NCL> SET DSA ACCOUNTING FACILITY ON
```

To disable accounting, enter the following command:

```
NCL> SET DSA ACCOUNTING FACILITY OFF
```

The value of the Accounting State attribute is maintained when you delete and re-create the DSA.

8.1.2. Configuring the Accounting Facility

There are five characteristic attributes of the DSA entity that you can use to configure the DSA accounting facility, and one read-only attribute that displays information about the accounting facility:

- The Accounting Options characteristic attribute controls whether the DSA includes the Protocol Data Unit (PDU) of user requests and errors in accounting records.
- The Accounting Rollover Interval characteristic attribute controls how frequently the DSA perform s a scheduled rollover of the accounting file.
- The Accounting Rollover Window characteristic attribute controls how long the DSA has got to perform a scheduled accounting file rollover before abandoning the attempt.
- The Accounting Rollover Start Time characteristic attribute controls when the first accounting file rollover takes place. Thereafter, the DSA perform s accounting file rollover according to the setting of the Accounting Rollover Interval attribute.
- The Last Accounting Rollover Time characteristic attribute indicates when the most recent accounting file rollover was performed. It is a read-only attribute.
- The Accounting Rollover Unscheduled Time characteristic attribute can be used to force an immediate unscheduled accounting file rollover, or a rollover at any given time.

For more detailed information on these characteristic attributes, see the Directory Module online help.

8.1.3. The Location and Filename of the Accounting File

The DSA creates a new accounting file when you enable accounting. Accounting files are stored in the directory pointed to by the DXD$ACCOUNTING logical. VSI recommends that you store these directories on a different disk to the one that contains the DSA's data files. To do so, define the

logical DXD$ACCOUNTING to point to a directory on a different disk to the one that contains DXD $DIRECTORY.

Each accounting file is named according to the following convention:

`dxd_YYYYMMDDhhmmss.dat`

where *YYYYMMDDhhmmss* is the date and time in universal time at which the accounting file was created, for example, *dxd_20001015122530.dat*. This accounting file was created at 12:25:30 on the 15th of October, 2000.

8.1.4. Managing Accounting File Rollover

Accounting file rollover is the process by which the accounting facility closes the current accounting file and creates a new one. You can then decode and process the closed accounting file using your decoding and processing utilities. Accounting file rollover can be either a scheduled accounting file rollover or an unscheduled accounting file rollover.

A scheduled accounting file rollover is one that is forced by the settings of the Accounting Rollover Start Time or Accounting Rollover Interval characteristic attributes of the DSA entity. You need to decide how often you want to process accounting files and set the DSA to perform scheduled accounting file rollovers accordingly.

An unscheduled accounting file rollover is one that is forced by the use of the Accounting Rollover Unscheduled Time characteristic attribute.

These attributes are described in more detail in the Directory Module online help.

Warning

Note that the closed accounting files are not deleted automatically; that is the responsibility of the accounting manager.

8.1.5. Backing Up Accounting Files

The accounting files can be backed up using the normal backup procedure. To ensure that the backup is up to date, either perform an unscheduled accounting file rollover or disable accounting before you back up the accounting file.

8.2. Processing the Accounting File

This section contains information needed by the person responsible for processing the accounting file. This information includes the ASN.1 definitions of the records the DSA writes to the accounting file.

Section 8.2.4 contains some important information that you need to be aware of in order to understand and decode the accounting file successfully.

8.2.1. Types of Record in the Accounting File

There are six types of record that can be written to the accounting file by the DSA. Each type is explained in more detail in Section 8.2.3.

- File Start record

A File Start record is the first record in an accounting file. It includes information about when the accounting file was started and, if this is not the first accounting file created by the DSA since the DSA was last created, the file name of the previous accounting file.

- Session Start record

  The accounting facility creates a session start record when a user connects to the DSA using the DXIM BIND command, and also when the DSA is contacted by another DSA. A Session Start record may contain for example, the distinguished name and authentication level of the user who started the session.

- Operation record

  An Operation record contains information about a user request and the DSA's response to the request, for example, whether the request was successful and, for read, search, and list operations, how much information was returned to the user.

- Session End record

  A Session End record contains, for example, information about how long a session lasted and why it ended.

- Discard record

  A Discard record indicates that the DSA has discarded some records without writing them to the accounting file. This only happens in exceptional circumstances, for example, when the DSA cannot access the accounting file for some reason. The Discard record contains information about how many records were discarded and indicates the time at which the first record was discarded.

- File End record

  A File End record is the last record in an accounting file. It contains information about when the accounting file was closed.

8.2.2. The ASN.1 Definition of Accounting Record Elements

This section describes the ASN.1 definition of the elements of information included in accounting records. Sections Section 8.2.3.1 to Section 8.2.3.6 describe the ASN.1 definitions of the different types of accounting record. The ASN.1 definitions of accounting records include the elements defined in this section.

Each record in the accounting file consists of a set of elements of information. The set is different for each different type of record.

The ASN.1 definitions of the elements are as follows:

Local Accounting Serial Number

The Local Accounting Serial Number element (LASN) uniquely identifies each accounting record within the accounting file. It consists of a `session-number` and `request-number`.

LASN is defined as follows:

```
LASN ::= SEQUENCE {
        session-number INTEGER,
        request-number INTEGER }
```

Directory Instance Identifier

The Directory Instance Identifier element (`DII`) uniquely identifies the DSA that created the accounting file.

`DII` is defined as follows:

```
DII ::= SEQUENCE {
        dsa-name            DistinguishedName,
        dsa-creation-stamp  GENERALIZED TIME}
```

- `dsa-name` is the distinguished name of the DSA that wrote the record. The distinguished name is defined in ITU-T Recommendation X.501.
- `dsa-creation-stamp` indicates the time (in generalized time format) at which the DSA that wrote the record was created. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.

Results Summary

The Results Summary element (`ResultsSummary`) gives brief information about the results of a read, search, or list operation. You can use the Accounting Options characteristic attribute (see the Directory Module online help for more information) to include more detailed information about the results of a user request in the Result PDU and Error PDU elements.

```
ResultsSummary ::= SEQUENCE {
        size     INTEGER,
        entries  INTEGER }
```

- `size` is the number of octets used in the results.
- `entries` is the number of entries included in the results.

Operation Status

The Operation Status element (`OperationStatus`) describes whether an operation succeeded or failed. If the operation failed, `OperationStatus` also describes the reason why it failed. For example, an `OperationStatus` with a value of 5 indicates that an operation failed because of a security problem.

`OperationStatus` is defined as follows:

```
OperationStatus ::= INTEGER {
        Success (1),
        Abandoned (2),
        RequestError (3),
        Referral (4),
        SecurityProblem (5),
        IncompleteResult (6),
        ServiceError (7) }
```

An `OperationStatus` of 2 to 7 indicates an error or problem. The following list indicates what X.500 errors are represented by each Operation Status code. Each error is defined in ITU-T Recommendation X.511.

- The `Abandoned` operation status is not currently returned by Enterprise Directory, and is included for future use only.
- `RequestError` indicates an X.500 Name error, Attribute error, or Update error.
- `Referral` indicates an X.500 Referral.
- `SecurityProblem` indicates an X.500 Security Error.
- `IncompleteResult` indicates either a Partial Outcome Qualifier, or an X.500 Service Error of Time or Administration Limit Exceeded.
- `ServiceError` indicates any X.500 Service Error other than Time or Administration Limit Exceeded.

If you need more information about the errors, set the Accounting Options characteristic attribute to include the ERRORPDU value.

If no `OperationStatus` is indicated in the record, the operation was a success.

Operation Type

The Operation Type element (`OperationType`) indicates the type of user request for which this accounting record was created. The operation types are defined in the ITU-T X.500 Recommendations.

`OperationType` is defined as follows:

```
OperationType ::= INTEGER {
        READ (1),
        COMPARE (2),
        ABANDON (3),
        LIST (4),
        SEARCH (5),
        ADD (6),
        REMOVE (7),
        MODIFY (8),
        MODIFYRDN (9) }
```

Session Status

The Session Status element (`SessionStatus`) describes why a session ended, and is defined as follows:

```
SessionStatus ::= ENUMERATED {
        Unbind (1),
        IdleDisconnect (2),
        TransportDisconnect (3) }
```

Protocol Type

The Protocol Type element (`ProtocolType`) indicates whether the connection was from a DUA or a DSA. A `ProtocolType` of `DAP` indicates that the connection is between a DUA and a DSA. A `ProtocolType` of `DSP` indicates that the connection is between two DSAs. If the protocol type is not specified in the record, the protocol of the connection is DAP.

`ProtocolType` is defined as follows:

```
ProtocolType ::= ENUMERATED {
```

```
        DAP (1),
        DSP (2) }
```

Authentication Level

The Authentication Level element (`AuthenticationLevel`) indicates the amount of authentication supplied by the user who made the request. If no authentication level element is specified in the record, the authentication level is none.

`AuthenticationLevel` is defined as follows:

```
AuthenticationLevel ::= ENUMERATED {
        none (0),
        simple (1),
        strong (2) }
```

Request PDU

The Request PDU element (`RequestPDU`) is the PDU of a user request. The structure of the PDU is defined in the ITU-T X.500 Recommendations. This information is only included if the Accounting Options characteristic attribute is set to include the request PDU. See the Directory Module online help for information about how to set this characteristic attribute.

`RequestPDU` is defined as follows:

```
RequestPDU ::= OCTETSTRING
```

The person responsible for managing the accounting facility needs to decide whether this information is necessary for the accounting policy. If the information is needed, the Accounting Options characteristic attribute needs to be set accordingly. Note that including this information increases the size of the accounting records and therefore means that the accounting file will consume extra disk space.

Error PDU

The Error PDU element (`ErrorPDU`) is included when the Operation Status of a user request is anything other than a success. The structure of the error PDU is defined in ITU-T Recommendation X.511. This information is only included if the Accounting Options characteristic attribute is set to include the error PDU. See the Directory Module online help for information on how to set this characteristic attribute.

`ErrorPDU` is defined as follows:

```
ErrorPDU ::= OCTETSTRING
```

The person responsible for managing the accounting facility needs to decide whether this information is necessary for the accounting policy. If the information is needed, the Accounting Options characteristic attribute needs to be set accordingly. Note that including this information increases the size of the accounting records and therefore means that the accounting file will consume extra disk space.

8.2.3. The Information Included in Accounting Records

Each record in the accounting file is identified by a context specific constructed tag, called the `AccountingRecord`. The ASN.1 definition of the `AccountingRecord` is as follows:

```
AccountingRecord ::= CHOICE {
        [0]SessionStartRecord,
```

```
        [1]SessionEndRecord,
        [2]OperationRecord,
        [3]FileStartRecord,
        [4]FileEndRecord,
        [5]DiscardRecord }
```

For example, each Session End record in the accounting file is given a context- specific constructed tag of [1].

The following subsections describe the ASN.1 definitions of each type of accounting record in the accounting file.

8.2.3.1. Session Start Record

A Session Start record indicates the start of a session between the DSA that created the accounting file and either a DUA or another DSA. The start of a session is indicated by a BIND operation. The Session Start record is always the first record written to the accounting file for each session. The Session Start record is followed by one or more Operation records and a Session End record.

The following is the ASN.1 definition of a Session Start record:

```
[0]SessionStartRecord ::= SEQUENCE {
        request-id      LASN,
        start-time      GENERALIZED TIME,
        calling_NSAP    OCTET STRING,
        protocol        ProtocolType DEFAULT {DAP},
        originator      DistinguishedName OPTIONAL,
        authentication  [0] AuthenticationLevel DEFAULT {none} }
```

- `request-id` is the Local Accounting Serial Number (LASN), consisting of a session number and a request number. The request number is always 1 for a Session Start record.
- `start-time` is the time at which the session was started. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.
- `calling_NSAP` is the NSAP of the DUA or DSA making the connection.
- `protocol` indicates whether the session is started by a DUA (protocol type DAP) or another DSA (protocol type DSP).
- `originator` is the distinguished name of the user making the request. The distinguished name is not included if the originator's authentication level is none. The format of a distinguished name is defined in ITU-T Recommendation X.501.
- `authentication` is the user's authentication level.

8.2.3.2. Session End Record

A Session End record indicates the end of a session. The Session End record is not always the last recor d relating to a session to be written to the accounting file, because the DSA writes accounting records asynchronously. Consequently, the Session End record in the accounting file can be followed by one or more Operation records relating to operations that occurred before the end of the session.

The ASN.1 definition of a Session End record is as follows:

```
[1]SessionEndRecord ::= SEQUENCE {
        request-id       LASN,
```

```
        start-time       GENERALIZED TIME,
        calling_NSAP     OCTET STRING,
        protocol         ProtocolType DEFAULT {DAP},
        originator       DistinguishedName OPTIONAL,
        authentication   [0] AuthenticationLevel DEFAULT {none},
        session-end-time GENERALIZED TIME,
        session-elapsed  INTEGER,
        session-status   SessionStatus }
```

- `request-id` is the Local Accounting Serial Number (LASN), consisting of a session number and a request number.
- `start-time` is the time at which the session was started. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.
- `calling_NSAP` is the NSAP of the DUA or DSA making the connection.
- `protocol` indicates whether the session is started by a DUA (protocol type DAP) or another DSA (protocol type DSP).
- `originator` is the distinguished name of the user making the request. The distinguished name is not included if the originator's authentication level is none. The format of a distinguished name is defined in ITU-T Recommendation X.501.
- `authentication` is the user's authentication level.
- `session-end-time` indicates when the session ended.
- `session-elapsed` indicates the total time, in milliseconds, of the session. This includes time in which the session was idle and also time spent by the DSA waiting for information from other DSAs.
- `session-status` indicates why the session ended.

8.2.3.3. Operation Record

An Operation record contains information about a user request. The DSA writes a unique Operation record for each user request.

The ASN.1 definition of an Operation record is as follows:

```
[2]OperationRecord ::= SEQUENCE {
        request-id       LASN,
        start-time       GENERALIZED TIME,
        elapsed          INTEGER,
        operation-type   OperationType,
        protocol         ProtocolType DEFAULT {DAP},
        originator       DistinguishedName OPTIONAL,
        authentication   [0]AuthenticationLevel OPTIONAL,
        status           [1]OperationStatus DEFAULT Success,
        results          [2]ResultsSummary OPTIONAL,
        request-pdu      [3]RequestPDU OPTIONAL,
        error-pdu        [4]ErrorPDU OPTIONAL }
```

- `request-id` is the Local Accounting Serial Number (LASN), consisting of a session number and a request number.
- `start-time` is the time at which the operation was started. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.

- `elapsed` indicates the total time in milliseconds that the DSA took to process the user request. This includes the time spent connecting to other DSAs and waiting for information from those DSAs.
- `operation-type` is the type of user request the Operation record is for.
- `protocol` indicates whether the session is started by a DUA (protocol type DAP) or another DSA (protocol type DSP).
- `originator` is the distinguished name of the user making the request. The distinguished name is not included if the originator's authentication level is none. The format of a distinguished name is defined in ITU-T Recommendation X.501.
- `authentication` is the user's authentication level.
- `status` describes whether an operation succeeded or failed. If the operation failed, status also describes the reason why it failed.
- `results` is a summary of the results of a read, search, or list operation sent back to the user who made the request. The summary includes the size of the returned results in octets and the number of entries included in the results.
- `request-pdu` is the PDU of the user request. It is only included if the setting of the Accounting Options characteristic attribute includes REQUESTPDU (see the Directory Module online help for more information).
- `error-pdu` is the error PDU for a failed user request. It is only included if the setting of the Accounting Options characteristic attribute includes ERRORPDU, and the status element has a value other than success (see the Directory Module online help for more information).

8.2.3.4. File Start Record

The DSA writes a File Start record when it starts a new accounting file. This can be either when you enable the accounting facility or when the DSA perform s accounting file rollover. The ASN.1 definition of a File Start record is as follows:

```
[3]FileStartRecord ::= SEQUENCE {
        dsa-instance        DII,
        start-time          GENERALIZED TIME,
        current-file        IA5String,
        previous-file       IA5String OPTIONAL }
```

- `dsa-instance` is a Directory Instance Identifier (DII) consisting of the DSA name and creation time stamp of the DSA that wrote the record.
- `start-time` is the time at which the new accounting file was started. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.
- `current-file` is the file specification of the accounting file that contains the record.
- `previous-file`, if present, is the file name of the previous accounting file. The `previous-file` is not included if this accounting file is the first accounting file this invocation of the DSA has created. For example, when accounting is first enabled, the `previous-file` is null.

8.2.3.5. File End Record

When the DSA closes an accounting file, it writes a File End record at the end of that accounting file, unless a problem prevents the DSA writing the File End Record. The last record in an accounting file

should therefore be a File End record, unless a problem prevents the DSA writing all outstanding records into the accounting file when it perform s accounting file rollover. The DSA may close an accounting file for any one of the following reasons:

- The DSA perform s accounting file rollover.
- Accounting is disabled.
- The DSA is deleted.

If the DSA is deleted, or if accounting is disabled, the DSA immediately writes all outstanding records to the accounting file. If the DSA cannot write one of the records into the accounting file, it discards all the remaining records. In this case, the DSA does not include a File End record.

The ASN.1 definition of a File End record is as follows:

```
[4]FileEndRecord ::= SEQUENCE {
        dsa-instance     DII,
        end-time         GENERALIZED TIME,
        current-file     IA5String }
```

- `dsa-instance` is a Directory Instance Identifier (DII) consisting of the DSA name and creation time stamp of the DSA that started the accounting file.
- `end-time` is the time at which the current accounting file was closed. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.
- `current-file` is the file specification of the accounting file that contains this record.

8.2.3.6. Discard Record

A Discard record is created when the DSA discards a record without writing it to the accounting file. This may happen either when the DSA cannot write records to the accounting file, for example, because the DSA does not have enough memory to store the record and write it to the accounting file, or because the DSA cannot open the accounting file for some reason. In all cases, the DSA issues an Accounting Records Discarded event, and continues its attempts to write records to the accounting file. When the DSA can access the accounting file again, it writes a Discard record to the accounting file that indicates how many records have been lost.

The ASN.1 definition of a Discard record is as follows:

```
[5]DiscardRecord ::= SEQUENCE {
        time              GENERALIZED TIME,
        discarded-count   INTEGER }
```

- `time` is the time at which the DSA discarded the first recor d. GENERALIZED TIME is defined in the ISO8824 ASN.1 standard.
- `discarded-count` is the number of records the DSA discarded without writing them to the accounting file.

8.2.4. Notes About Accounting Files

This section contains information about accounting files that you need to be aware of:

- Accounting is considered a non-critical function. If the DSA fails to write records to the accounting file for any reason, the DSA will continue to process user requests but will not be able to write

any accounting information to the accounting file. When the problem that is preventing the DSA recording accounting information is solved, the DSA writes a Discard record to the accounting file, as described in Section 8.2.1, and generates an Accounting Records Discarded event.

- Accounting records cannot be guaranteed to be written to the accounting file in chronological order. For example, the record for Session 1, Request 3 may be written to the accounting file before the record for Session 1, Request 2. Similarly, a Session End record can be written before one or more Operation records relating to that session. However, the Session Start recor d is always the first accounting recor d written for a session.

  Your decoding utility needs to be able to cope with the fact that accounting records will not always be in chronological order.

- An accounting record is created when the DSA has processed a user request and has returned the results of that request to the user. This means that the DSA will write an operation record to the accounting file even if the user does not receive the results of the request due to a system or network failure.

- The elapsed time recorded in Operation records in the accounting file is the time taken by the DSA to completely process a user request. If the DSA has to contact a second DSA in order to process a user request, the time taken by this connection is included in the elapsed time in the Operation record.

  However, the second DSA contacted will also write an Operation record to its accounting file, if accounting is enabled on that DSA. This Operation record will have a protocol type of DSP, and will indicate the elapsed time taken by the second DSA to process the request from the original DSA.

  This means that the time taken by the DSP connection is included in Operation records in two separate accounting files. For example, DSA1 processes a user request. After 5 seconds DSA1 contacts DSA2. It takes DSA2 10 seconds to return the required information to DSA1. DSA1 then returns the information to the user. If the accounting facility is enabled on both DSA1 and DSA2, each DSA writes an Operation record for this user request. DSA1 writes an Operation record with a protocol type of DAP and an elapsed time of 15 seconds. DSA2 writes an Operation record with a protocol type of DSP and an elapsed time of 10 seconds.

  You need to be aware of this when you bill the user who made the request.

Chapter 9. Error Messages

This chapter lists Directory Service error messages, describes the cause of the errors, and explains how to recover from them.

- Section 9.1 lists errors issued by NCL when handling Directory Service entities.
- Section 9.2 lists the errors issued by DXIM.

9.1. NCL Messages

This section describes the messages returned by NCL when handling Directory Service entities. For standard NCL responses, refer to the appropriate NCL documentation.

The messages are listed in alphabetical order. The following references provide easy access to specific errors. Some of the references are abbreviated.

Alias entry prevents creation (see Alias Entry Prevents Creation)
Alias entry prevents deletion (see Alias Entry Prevents Deletion)
A Superior Naming Context ... (see A Superior Naming Context that is Not Correctly Terminated by a Subordinate Reference Prevents Creation)
A Superior Shadow Naming Context ... (see A Superior Shadow Naming Context that is Not Correctly Terminated by a Subordinate Reference Prevents Creation)
Cannot create a Naming Context ... (see Cannot Create a Naming Context at the Root of the DIT)
Cannot create a Subordinate Ref. at the root ... (see Cannot Create a Subordinate Reference at the Root of the DIT)
Cannot create a Subordinate Ref. in a shadow ... (see Cannot Create a Subordinate Reference in a Shadow Naming Context)
Cannot read address for specified DSA (see Cannot Read Address for Specified DSA)
Failed in performing update ... (see Failed in Performing the Update due to Insufficient Resources, DSA Deleted)
Invalid data received from supplier (see Invalid Data Received from Supplier)
Invalid data received from supplier, DSA deleted (see Invalid Data Received from Supplier, DSA Deleted)
Schema Warning: The memory image file ... (see Schema Warning: The memory image file does not use the current schema)
Supplied update incompatible with the DSA (see Supplied Update Incompatible with the DSA)
The alias entry with the same name ... (see The Alias Entry with the Same Name Must be Deleted Before the Naming Context can be Deleted)
The directly superior entity ... (see The Directly Superior Entity is Another Subordinate Reference)
The DSA already holds an alias entry ... (see The DSA Already Holds an Alias Entry with the Specified Name)
The DSA already holds a Naming Context ... (see The DSA Already Holds a Naming Context Entity with the Specified Name)
The DSA already holds an entry ... (see The DSA Already Holds an Entry with the Specified Name)
The DSA already holds entries or entities ... (see The DSA Already Holds Entries or Entities Subordinate to the Entity Being Created)
The DSA entity already exists (see The DSA Entity Already Exists)
The DSA entity is not in the correct state (see The DSA Entity is Not in the Correct State)
The DSA holds entries or entities subordinate ... (see The DSA Holds Entries or Entities Subordinate to the Entity Being Deleted)

The DSA information tree is corrupt (see The DSA Information Tree is Corrupt)
The DSA information tree is incompatible (see The DSA Information Tree is Incompatible with This Version of the DSA)
The DSA is currently being created (see The DSA is Currently Being Created)
The DSAs AE title attribute has not been set (see The DSAs AE Title Attribute Has Not Been Set)
The DSAs presentation address attribute ... (see The DSAs Presentation Address Attribute Has Not Been Set)
The entity name is not a valid directory name (see The Entity Name is Not a Valid Directory Name)
The license check has failed for this product (see The License Check has Failed for this Product)
The Naming Context entity with the same name ... (see The Naming Context Entity with the Same Name Must be Deleted First)
There are insufficient resources to perform ... (see There are Insufficient Resources to Perform the Update)
The schema is corrupt (see The Schema is Corrupt)
The schema is incompatible with this version ... (see The Schema is Incompatible with This Version of the DSA)
The Shadow Naming Context with the same name ... (see The Shadow Naming Context with the Same Name Must be Removed First)
The Superior Reference entity must be removed first (see The Superior Reference Entity Already Exists)
The Supplier argument is not a valid directory name (see The Supplier Argument is Not a Valid Directory Name)
The Supplier argument is not a valid pres. address (see The Supplier Argument is Not a Valid Presentation Address)
The supplier DSA is not available (see The Supplier DSA is Unavailable)
The supplier DSA rejected the connection request (see The Supplier DSA Rejected the Connection Request)
The update failed due to a communications problem (see The Update Failed Due to a Communications Problem)
This Accessor entity already exists (see This Accessor Entity Already Exists)
This Naming Context entity already exists (see This Naming Context Entity Already Exists)
This Subordinate Reference entity already exists (see This Subordinate Reference Entity Already Exists)
Unexpected failure (see Unexpected Failure)
Unrecognized command: Verb not permitted ... (see Unrecognized command: Verb not permitted for this entity class)
You cannot delete a Naming Context ... (see You Cannot Delete a Naming Context That Contains Entries)
You cannot delete a shadow Naming Context (see You Cannot Delete a Shadow Naming Context)
You cannot delete a shadow Subordinate Reference (see You Cannot Delete a Shadow Subordinate Reference)

Alias Entry Prevents Creation

The command you have specified contains an alias. The DSA does not support the use of alias names when managing entities. Specify the distinguished name of the entry.

Alias Entry Prevents Deletion

The command you have specified contains an alias. The DSA does not support the use of alias names when managing entities. Specify the distinguished name of the entry.

A Superior Naming Context that is Not Correctly Terminated by a Subordinate Reference Prevents Creation

You cannot create a Naming Context entity because a superior naming context has no terminating Subordinate Reference entity. The name of the superior Naming Context entity is also returned.

Create a Subordinate Reference entity to terminate the superior naming context, and then create the new Naming Context entity.

A Superior Shadow Naming Context that is Not Correctly Terminated by a Subordinate Reference Prevents Creation

You cannot create a Naming Context entity because a superior shadow naming context has no terminating Subordinate Reference entity. The names of the incomplete shadow Naming Context entity and its master DSA are also returned.

At the master DSA, create a Subordinate Reference entity to terminate the superior Naming Context entity. Retry the sequence of updates to create the shadow Naming Context entity. Then create the new Naming Context entity. See Section 4.1.4 and Section 4.1.5 for more details.

Cannot Create a Naming Context at the Root of the DIT

You cannot create a Naming Context entity directly at the root of the DIT. A Naming Context entity must have a distinguished name that contains at least one relative distinguished name.

Refer to *VSI Enterprise Directory Management* for more information on how to plan Naming Context entities.

Cannot Create a Subordinate Reference at the Root of the DIT

You cannot create a Subordinate Reference entity directly at the root of the DIT. A Subordinate Reference entity must have a distinguished name that contains at least one relative distinguished name.

Refer to *VSI Enterprise Directory Management* for more information on how to plan Subordinate Reference entities.

Cannot Create a Subordinate Reference in a Shadow Naming Context

You cannot create a Subordinate Reference entity to terminate a shadow Naming Context entity, because the shadow DSA does not own that part of the DIT.

Create the Subordinate Reference entity on the master DSA, and then use replication to provide the shadow DSA with a copy of the Subordinate Reference entity.

Cannot Read Address for Specified DSA

The DSA cannot read the address of the supplier DSA whose AE title is specified in the UPDATE directive. The entry in the DIT representing the supplier DSA is not accessible. Retry the command specifying the Presentation Address of the supplier DSA instead of its AE title.

Check for the existence of a directory entry representing the supplier DSA. If the entry does not exist, refer to *HP Enterprise Directory — Management* for details of planning entries to represent your DSAs. If the entry exists, but is not accessible to the local DSA, it may be that your DSAs do not trust each other sufficiently for the local DSA to read information held by another DSA, or it may be that the local DSA is configured not to chain requests to other DSAs.

If you follow the VSI advice, all DSAs have local copies of the entries representing all other DSAs. This means that the consumer DSA should be able to read the address of the supplier DSA from its local database. If this is not happening, refer to *HP Enterprise Directory — Management* for advice about how to replicate the DSA entries to all DSAs.

Failed in Performing the Update due to Insufficient Resources, DSA Deleted

Insufficient resources are available to complete the update. This means that the consumer DSA has insufficient disk space or insufficient memory to write the update log to disk before applying it to its database. Because the DSA has already made changes to its database. It therefore deletes itself to prevent corruption of its existing data. When you recreate the DSA, it recovers the data that it held prior to the failed update.

In addition to this error message, NCL generates an event containing reason information indicating the resource that is missing. Using this reason information, see Section 7.3 for recovery information.

Invalid Data Received from Supplier

The update supplied contains invalid data. This usually means that there is a problem with the protocol passed between the supplier DSA and consumer DSA. The consumer DSA therefore does not apply any changes to its database. Try the update again, to see whether the error recurs.

Invalid Data Received from Supplier, DSA Deleted

The update supplied contains invalid data. This usually means that there is a problem with the protocol passed between the supplier DSA and consumer DSA. However, the consumer DSA has already started applying changes to its database before detecting the error. It therefore deletes itself to prevent corruption of its data. When you recreate the consumer DSA, it recovers the database that it held prior to the failed update.

Supplied Update Incompatible with the DSA

The information from the supplier DSA is incompatible with the schema held by the consumer DSA. After this has occurred, the consumer DSA is automatically deleted to ensure that the consumer DSA's DIB fragment is not corrupted by the attempted replication operation.

See Section 4.5.7 for recovery procedure.

The Alias Entry with the Same Name Must be Deleted Before the Naming Context can be Deleted

The Naming Context entity coexists with an alias entry. You cannot delete the Naming Context entity until you have deleted the alias entry.

Use DXIM to delete the alias entry and then reissue your command.

The Directly Superior Entity is Another Subordinate Reference

A superior Subordinate Reference entity exists. You cannot create consecutive Subordinate Reference entities.

No action is necessary.

The DSA Already Holds an Alias Entry with the Specified Name

This error can be returned under two circumstances:

- When creating a Subordinate Reference entity

  An alias entry with the same name already exists within a naming context. This Subordinate Reference entity is unnecessary.

- When creating a Naming Context entity

  An alias entry with the same name already exists. The alias entry must already be within a naming context. You cannot create a naming context where an alias entry already exists. If you wish to continue and create the Naming Context entity, delete the existing alias entry, create a Subordinate Reference entity and then create the Naming Context entity.

The DSA Already Holds a Naming Context Entity with the Specified Name

A Naming Context entity already exists at this position in the DIT. You cannot create a Subordinate Reference entity if a Naming Context entity already exists at that location, since this would result in a naming context starting and terminating at the same point.

The DSA Already Holds an Entry with the Specified Name

This error can be returned under two circumstances:

- When creating a Subordinate Reference entity

  An entry with the name you have specified already exists as part of a naming context. Therefore, there is no need for a Subordinate Reference entity. No action is necessary.

- When creating a Naming Context entity

An entry with the name you have specified already exists as part of a naming context. You cannot create a Naming Context entity if a directory entry with the same name already exists. If you wish to continue and create the Naming Context entity, delete the existing directory entry, create a Subordinate Reference entity and then create the Naming Context entity.

The DSA Already Holds Entries or Entities Subordinate to the Entity Being Created

This error can be returned under two circumstances:

- When creating a Naming Context entity

  You cannot create a Naming Context entity at this position in the DIT because the name you have specified has subordinates. You must create Naming Contexts in a hierarchical order (top down) according to the DIT structure rules. Refer to *VSI Enterprise Directory Management* for more information.

- When creating a Subordinate Reference entity

  Subordinate entries exist below the proposed location of the Subordinate Reference entity. To create a Subordinate Reference entity in the proposed position would leave these entries beyond the termination point of whichever naming context they are part of.

The DSA Entity Already Exists

The DSA entity you tried to create already exists. No action is necessary.

The DSA Entity is Not in the Correct State

The DSA is in the wrong state for the operation to be performed. The current state of the DSA is returned with the error message. The following rules apply:

- The UPDATE and DISABLE directives can be used only when the DSA is in state ON.
- The DELETE and ENABLE directives can be used only when the DSA is in state OFF.
- The ADD and REMOVE directives can be used when the DSA is in state ON or state OFF.
- The SHOW directive can be used when the DSA is in states ON, ENABLING, DISABLING, or OFF.
- The SET directive can be used when the DSA is in states ON, ENABLING, DISABLING or OFF to modify all attributes except for:
  - AE Title
  - Presentation Address
  - Volatile Modifications
  - Schema Check on Modify

  To modify these attributes, the DSA must be in state OFF.

Check the state of the DSA using the SHOW STATE directive. Change the state if necessary, or wait for outstanding directives that change the DSA state to complete, then retry your command.

The DSA Holds Entries or Entities Subordinate to the Entity Being Deleted

This error can be returned under two circumstances:

- When deleting a Naming Context entity

  You cannot delete a Naming Context entity at this position in the DIT because the name you have specified has subordinates. You must delete the subordinate entries or entities before you can delete the Naming Context. Refer to *VSI Enterprise Directory Management* for more information.

- When deleting a Subordinate Reference entity

  The Subordinate Reference entity has subordinates and therefore cannot be deleted. You must delete the subordinate entries or entities before you delete the Subordinate Reference entity.

The DSA Information Tree is Corrupt

The copy of the DSA Information Tree stored on disk is corrupt and consequently not loaded into memory.

Delete the DSA database files and recreate the DSA. Ensure that the DSA attribute Schema Check on Modify is set to TRUE and repopulate the DSA as described in *VSI Enterprise Directory Management*. Refer to Section 2.4 for more information.

The DSA Information Tree is Incompatible with This Version of the DSA

The copy of the DSA information tree stored on disk is incompatible with the version of the DSA. Normally this is due to the version of the database on disk being from an older version of the DSA.

Ensure that the DSA attribute Schema Check on Modify is set to TRUE. Delete the DIB fragment and recreate the DSA. Repopulate the DSA as described in *VSI Enterprise Directory Management*. Refer to Section 2.4 for more information.

The DSA is Currently Being Created

A DSA with the name you specified is already being created. No action is necessary.

The DSAs AE Title Attribute Has Not Been Set

The DSA entity you are trying to enable has no AE Title attribute defined. You must define an AE Title attribute for the DSA entity before you can enable it.

The DSAs Presentation Address Attribute Has Not Been Set

The DSA entity you are trying to enable does not have a Presentation Address attribute defined. You must define a valid Presentation Address for the DSA entity before you can enable it.

The Entity Name is Not a Valid Directory Name

The distinguished name you specified is not a valid distinguished name. The incorrect name is returned with the error message.

Enter a valid distinguished name.

The License Check has Failed for this Product

You tried to create a DSA, but the directive has failed because the DSA license is not registered or loaded.

Check to see whether the DSA license is registered and loaded, using the following command:

```
$ LICENSE LIST
```

The DSA license is called X500-DIRECTORY-S ERVER.

Load the DSA license. If you do not have a license, see the *Software Product Description* for information about ordering licenses.

The Naming Context Entity with the Same Name Must be Deleted First

You cannot delete the Subordinate Reference entity because a Naming Context entity coexists with the Subordinate Reference entity. To delete the Subordinate Reference entity would leave the DSA's knowledge information invalid.

You must first delete the Naming Context entity before you can delete the Subordinate Reference entity.

There are Insufficient Resources to Perform the Update

Insufficient resources are available to complete the update. This usually means that the consumer DSA has insufficient diskspace to write the shadow information to disk before applying it to its database. It can also mean that the consumer DSA has insufficient memory to add the shadow information to whatever information it already holds.

In addition to this error message, NCL generates an event containing reason information indicating the resource that is missing. Using this reason information, see Section 7.3 for recovery information.

The Schema is Corrupt

Check that the compiled schema file DXD$SCHEMA.DAT exists and is in the correct directory, DXD $DIRECTORY. If the schema file does exist and is in the correct directory but you get this error, it means that the file is corrupt. Therefore, edit the schema text files as necessary, recompile the schema and retry the command. Refer to Section 2.4 for more information.

The Schema is Incompatible with This Version of the DSA

The schema has been changed such that it is not compatible with the current database. Either change the schema so that it is compatible, or delete and recreate the database.

The Shadow Naming Context with the Same Name Must be Removed First

You cannot delete the Subordinate Reference entity because a shadow Naming Context entity coexists with the Subordinate Reference entity. To delete the Subordinate Reference entity would leave the DSA's knowledge information invalid.

You should only delete a Subordinate Reference entity if the Naming Context entity to which it refers is deleted. If you want to delete the Subordinate Reference entity, changing the DSA's knowledge information, complete the following steps:

1. Use the information returned with the NCL error to identify the master DSA.
2. Modify the consumer information at the master DSA, so that the shadow DSA is no longer listed as a supplier.
3. Update the shadow DSA. This removes the shadow Naming Context entity.
4. Delete the Subordinate Reference entity at the master node.

The Superior Reference Entity Already Exists

The Superior Reference entity already exists. No action is necessary.

The Supplier Argument is Not a Valid Directory Name

Check that the distinguished name you specified for the supplier DSA is valid.

The Supplier Argument is Not a Valid Presentation Address

The presentation address supplied for the supplier DSA is not valid. Check the format of the presentation address is valid.

The Supplier DSA is Unavailable

This error is returned in the following circumstances:

- The supplier DSA cannot verify the identity of the consumer DSA. See *VSI Enterprise Directory Management* for details of replication, and its prerequisite tasks.
- The error can also mean that the supplier DSA is unavailable temporarily, and a later attempt to replicate will succeed.
- The supplier DSA has insufficient disk space to write the shadow information to disk before sending it to the consumer DSA.

Check the events generated on the supplier DSA to see whether they explain the failure of replication.

The Supplier DSA Rejected the Connection Request

The supplier DSA is not ready for communication. Check the state of the supplier DSA. It must be in state ON for communication.

The Update Failed Due to a Communications Problem

Use the event information to identify the communications problem. See Chapter 3 for more information about solving communications problems.

This Accessor Entity Already Exists

You tried to create an Accessor entity when one of the same name already exists.

No action is necessary.

This Naming Context Entity Already Exists

You tried to create a Naming Context entity when one of the same name already exists.

No action is necessary.

This Subordinate Reference Entity Already Exists

You tried to create a Subordinate Reference entity when one of the same name already exists.

No action is necessary.

Unexpected Failure

An unexpected failure occurred when updating the DSA. Check the Shadow Update Failure event for more information.

Unrecognized command: Verb not permitted for this entity class

NCL does not recognize the DSA entity. No NCL commands for the DSA will work. The NCL dictionary file containing the necessary NCL definitions has been lost, probably due to a DECnet upgrade.

Reinstall the Base component of the Directory Service software and reboot the system

You Cannot Delete a Naming Context That Contains Entries

The naming context entity coexists with a directory entry. You cannot delete the Naming Context entity until you have deleted the directory entry.

Use DXIM to delete the directory entry and then retry your command.

You Cannot Delete a Shadow Naming Context

This error can be returned if you attempt to delete a shadow Naming Context entity from a V1.* DSA.

To delete a shadow Naming Context entity from a V1.* DSA, complete the following steps:

1. Use the information returned with the error message to identify the supplier DSA of the naming context.

2. At the supplier DSA, modify the consumer information such that the naming context is not replicated to this shadow DSA.

3. Update the shadow DSA. This removes the shadow naming context.

V2.0 DSAs do allow you to delete shadow naming contexts directly, using the DELETE DSA NAMING CONTEXT command. However, if the naming context was received from another V2.0 DSA, then it may be recreated automatically. It is therefore best to remove the consumer information from the supplier DSA, as described above.

The DELETE DSA NAMING CONTEXT command is intended for deleting naming context s from their master DSAs, and for deleting shadow naming contexts for which the supplier has become unavailable, such that the tasks listed above cannot be achieved.

You Cannot Delete a Shadow Subordinate Reference

The Subordinate Reference entity is a shadow copy that has been created through replication. The DSA does not own this Subordinate Reference entity and therefore cannot delete it.

To delete this Subordinate Reference entity, you must delete the Subordinate Reference entity on the master DSA and then retry the replication process. Do this as follows:

1. Use the information returned with the NCL error to identify the master DSA.

2. At the master DSA, modify the consumer information so that the Subordinate Reference entity is not replicated to this shadow DSA.

3. Update the shadow DSA. This removes the shadow Subordinate Reference entity.

Note that you should only delete a Subordinate Reference entity if you are changing your Directory Service configuration, or as a temporary measure.

See *VSI Enterprise Directory Management* for information about configuring a Directory Service.

9.2. DXIM Error Messages

This section contains a list of the error messages returned by the VSI Enterprise Directory administration facility (DXIM), and where possible includes advice about how to solve the problem that causes the error. Note that this list does not include all informational messages returned, or error messages that are self-explanatory.

The errors are listed in alphabetical order.

A Communications Error Occurred

Refer to Section 3.1 and Section 3.2.

A Name Form Defined in the Schema is Not Referenced by any Structure Rules

A required Structure Rule is missing from the schema, or the name form is incorrectly defined in the schema. See *VSI Enterprise Directory Management* for information about modifying the schema.

An Unspecified Error Has Been Reported From the DXD_UI Interface

Contact VSI.

Attribute Syntax Not Defined in Local Schema

Check that the user specified the attribute value correctly.

If the value is specified correctly, this is probably an interworking problem. The attribute definition in the schema is using a syntax that is not known to DXIM.

Cannot Setup Context Object

Contact VSI.

Communications Error: DXIM is Unable to Receive Data from the DSA

The connection between the DUA and the DSA has been lost, or could not be established. Refer to Section 3.1 and Section 3.2.

Communications Error: DXIM is Unable to Transmit Data to the DSA

The connection between the DUA and the DSA has been lost, or could not be established. Refer to Section 3.1 and Section 3.2.

Communications Error: The DSA Rejected Association Establishment

A remote DSA has rejected the attempts by the DUA to establish a connection. This is normally due to the user having insufficient or incorrect authentication parameters.

Communications Error: Association With the DSA Has Been Aborted

The association with the DSA has been aborted. This could be due to, for example, the connection being lost because of network problems.

Retry the Bind command. If this fails, refer to Section 3.1 or try connecting to a different DSA.

Communications Error: DXIM is Unable to Communicate with the DSA

DXIM has been unable to communicate with the DSA. The normal reason for this is that the DSA does not exist or is in the wrong state. Refer to Section 3.1.

Communications Error: DXIM is Unable to Decode the Response From the DSA

DXIM has received a response from a DSA that it cannot decode. The response is rejected by the DXIM.

If the response is a referral, it may be that protocol extensions are being used. These extensions are used if the network type is specified when DXIM's presentation address is entered. Modify the presentation address so that the network type is not specified.

Retry the command. If the problem persists, contact VSI.

Communications Error: No Memory

There is insufficient memory available to DXIM.

Refer to Section 7.3.2 for information on how to recover from memory-related problems.

Communications Error: ROSE Invoke Problem

The DSA has experienced a problem invoking a ROSE request, for example, it does not recognize one of the request arguments, or does not recognize the operation being requested.

See Section 3.4 for information about handling ROSE errors.

Communications Error: ROSE Reject

The DUA or the DSA has rejected a ROSE request. This may be due to, for example, unrecognized or badly structured protocol data units.

See Section 3.4 for information about handling ROSE errors.

Communications Error: The DSA has Returned a Bad Invoke ID

The DSA has rejected a request on the grounds that it has an invalid invoke identifier. Check that the user is still bound to the DSA, and, if necessary, reissue the Bind request. Retry the command that caused the error.

Communications Error: Unexpected Communications Event

The DUA has detected an unexpected communications event which is affecting the connection to the DSA.

Try again later or try connecting to a different DSA. The event log might contain information indicating the source of the problem. If the problem persists, contact VSI.

Continuation Reference

A continuation reference can be displayed for two reasons:

- An attempt to chain a request from one DSA to another has failed.

For example, if a DSA fails to bind to another DSA due to authentication problems or network problems, or because chaining is prohibited, the DSA returns a continuation reference instead.

- A request exceeds a limit set by a service control.

  For example, if a request involves many DSAs, but a time limit is exceeded, the DSAs can return continuation references.

A continuation reference contains the presentation address of the DSA. A given request may return more than one continuation reference. In some cases, the continuation reference is accompanied by partial results. These are the results that were gathered before the limit was exceeded, or despite the failure to connect to another DSA.

If you receive a continuation reference, and you are not satisfied by the results you received, if any, then you can attempt to reconfigure the limit that was exceeded, or you can attempt to use the information in the continuation reference to bind directly to the DSA it describes.

Refer to Section 5.2 for details of how to deal with a continuation reference.

Diagnostic information follows

The diagnostic information, if any, is a continuation reference. The continuation reference details the name and network address of a DSA that could not be contacted by the DSA to which DXIM is connected.

Section 5.2 provides details of how to deal with continuation references.

DSA Error: The Attribute *attribute* has Invalid Syntax

Attribute Error: Invalid Attribute Syntax

This may be returned under two circumstances:

- The value supplied for the specified attribute does not conform to the attribute syntax for an attribute of that type. Check that you specified the attribute correctly.
- The directory schema used by DXIM is not compatible with that being used by the DSA. If the schemas are not compatible, make any modifications necessary (see *VSI Enterprise Directory Management* for information about modifying the schema).

DSA Error: The Attribute Value *attribute = value* has Invalid Syntax

Attribute Error: Invalid Attribute Syntax

This may be returned under two circumstances:

- The value supplied for the specified attribute does not conform to the attribute syntax for an attribute of that type. Check that the user has specified the attribute correctly.
- The directory schema used by DXIM is not compatible with that being used by the DSA. If the schemas are not compatible, make any modifications necessary (see *VSI Enterprise Directory Management* for information about modifying the schema).

DSA error: The Attribute Type *attribute* is Undefined

DSA error: The Attribute Type *attribute=value* is Undefined

Attribute Error: Undefined Attribute Type

An undefined attribute type was provided as an argument to the Create, Remove, Modify or Rename operation.

DSA Error: Administrative Limit Exceeded Before Operation Could be Completed

Service Error: Administrative Limit Exceeded

The operation could not be completed within the administrative constraints defined by the directory (for example, time limit or number of results returned). Any results obtained before the administrative limit was reached are returned.

If this error occurs when you are trying to create a new entry, it can mean that the DSA entry limit has been exceeded. In this case, a Resource Exhausted event is generated by the DSA stating Insufficient License Capacity. You must reduce the entry count of the relevant DSA, or provide more licenses.

DSA Error: Alias Entry *entry* Detected Where Alias Entries are Not Permitted

Name Error: Alias Dereferencing Problem

The name specified by the user is an alias name. The error shows the name of the first alias entry encountered when trying to find the name specified by the user.

If the user was attempting a modification, either tell the user to use a distinguished name for the target entry, or reconfigure the master DSA that holds that entry to permit the use of aliases in modification operations (CREATE, DELETE, MODIFY, RENAME, and SET commands).

To set the characteristic attribute for the DSA, use the following NCL command on that DSA's system:

```
NCL> SET DSA DEREFERENCE ALIAS ON MODIFY TRUE
```

If the user specified the Dont Dereference Aliases control, tell them to try the command again without the control. Check that the DUA defaults and the DXIM defaults do not specify Dont Dereference Aliases. If they do, amend the defaults, or tell the user to use the Dereference Aliases control on their command.

If none of the above solves the problem, use the DXIM SHOW command to show the attributes of the alias entry specified in the error message, as follows:

```
dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIASES
```

where *name* is the name of the entry.

The `aliasedObjectName` attribute of the alias entry contains the name of an entry. If you use another SHOW command to display that entry, you should find that it is also an alias entry. Amend the first alias entry so that its `aliasedObjectName` attribute contains the same name as the `aliasedObjectName` attribute of the second alias entry. In this way, the next time the user tries their original command, it should succeed.

DSA Error: Alias Points to Nonexistent Entry Matched the Following Part of Entry Name

Name Error: Alias Problem

An alias has been dereferenced, but no target entry has been foun d.

Use the information returned with the error message to identify the faulty alias, and correct the definition. Note that when you create an alias entry, there is no check to ensure that the target entry of that alias actually exists. Therefore, an alias can point to an entry that does not exist.

DSA Error: Attribute *attribute type* Already Exists

Attribute Error: Attribute Or Value Already Exists This is returned under three circumstances:

- The attribute has been specified more than once in the command.

  Check the syntax of the command line to ensure that the attribute has been specified only once.

- The attribute already exists in the specified entry.

  Check that the attribute actually exists, by displaying all attributes of the entry as follows:

  ```
  dxim> SHOW ENTRY name ALL ATTRIBUTES MASTER INFORMATION
  ```

  where *name* is the name of the entry the user was trying to modify.

  Check the access control defined for the attribute. This may be preventing the user seeing or adding the attribute. See Chapter 6 for information about access control.

  If the attribute already exists, another attribute of that type cannot be added to the entry. If, however, the intention of the user was to add a new attribute value, then they must use the DXIM MODIFY ENTRY ADD VALUE command.

  If the attribute does not exist, retry the command. If the command continues to fail, it might be that the user has insufficient access controls to display or manipulate the specified attribute.

- The user does not have sufficient access control rights to display or modify the specified attribute.

  Access rights determine your ability to access and manipulate certain directory information. The fact that the user has insufficient access rights to add a new attribute may be expected system behavior.

  Check the attribute access control information as described in Section 6.1.1.

  Check the user's access rights as follows:

  - Check whether or not the user is bound to the DSA through an authenticated bind. Use the DXIM SHOW BINDING command to display information about the how the user is bound to the DSA. If the display includes authentication information, the user is authenticated to the DSA.

    If the DSA to which the user is bound does not recognize the user, authentication fails and the bind is rejected.

    Check for this situation by examining the DSA's event log for an Authentication Failure event, with the reason code Unrecognized User. Refer to Section 10.1.16 for more information on this event.

If the user has not performed an authenticated bind, ask them to perform an authenticated bind and retry the original command.

- Check that the request is not being chained. If the request is not being chained, it might be because the authentication level cannot be maintained.

  By default, a DSA will only chain a request to another DSA if it can maintain the same level of authentication. If, at any point, the DSA cannot chain the request because it cannot maintain authentication, then it returns a referral in response to your command.

  Using the DXIM command line interface, retry the original command twice; the first with the PREFER CHAINING service control specified and the second with the ALLOW CHAINING service control specified.

  If both commands complete with the same error as the original command specified by the user, then chaining and maintaining authentication while chaining is not the problem.

  If the first command completes with the same error message as the original command and the second command returns a referral, DSA trust between the two DSAs has not been established. Take corrective action as described in Section 6.1.2.

DSA Error: Attribute *attribute* Already Has Value *value*

Attribute Error: Attribute Or Value Already Exists

This may be returned under two circumstances:

- The attribute value has been specified more than once.

  Check the syntax of the command line, or the information in the Modify window, to ensure that the attribute value has been specified only once. If multiple values are specified, ensure that the values are syntactically different under the appropriate matching rule. For example, the matching rule for telephone numbers means that `telephoneNumber= "1 2 3"` is the same as `telephoneNumber = "123"`.

- The attribute value already exists for the specified attribute.

  Check whether the attribute value actually exists, by displaying all attributes of the entry as follows:

  ```
  dxim> SHOW ENTRY name ALL ATTRIBUTES MASTER INFORMATION
  ```

  where *name* is the name of the entry the user is trying to modify.

  If the attribute value does not exist, retry the command. If the command continues to fail, it may be that the user has insufficient access controls to display or manipulate the specified attribute value, so is unaware that the value already exists. Bind to the DSA as a manager and check the entry again.

DSA Error: Attribute Syntax Error in Entry Name Matched the Following Part of the Entry Name

Name Error: Invalid Attribute Syntax

This error may be returned under the following circumstances:

- An attribute value in the entry's distinguished name has a syntax that is incompatible with the attribute type.

  Each attribute value of a directory entry has an associated attribute syntax, for example, the common name attribute has String syntax.

  The DXIM online help contains a list of all supported directory attributes and their associated attribute syntaxes. Use this help to check the attribute syntax of each attribute in the entry's distinguished name. If necessary, amend the distinguished name and retry the command.

  For more information on attribute types, attribute syntaxes and the directory schema, refer to *VSI Enterprise Directory Management*.

- An attribute value violates a constraint on the attribute type.

  Most attributes have a constraint defined that determines the values that the attribute can have, for example, an attribute value must be within a given range of values.

  The DXIM online help contains a list of all supported directory attributes and defines the constraints imposed on the values that those attributes can have. Use this help to check that each distinguished value in the command adheres to the constraints imposed by the attribute's definition.

  If necessary, amend the distinguished value and retry the command.

  For more information on attributes, attribute definitions and attribute constraints, refer to *VSI Enterprise Directory Management*.

- No naming value is supplied for an entry being created using the Create window.

DSA Error: Constraint Violation in Attribute *attribute type*

Attribute Error: Constraint Violation

This may be returned under three circumstances:

- An attribute value does not meet the constraints imposed by the attribute definition in the directory schema. For example, the attribute value is outside a range of values, or it may exceed the maximum size allowed.

  Check the value the user has entered for the specified attribute against the attribute definition, as described by the DXIM online help (or as defined in the schema source if you have customized the schema). The attribute definition defines whether any size restrictions apply to the attribute, or whether its value must be within a specified range of values.

  Correct the attribute as appropriate and retry the command.

- Multiple attribute values are being added to a single-valued attribute.

  If the user is adding multiple values, refer to the DXIM online help, use the DXIM SHOW SCHEMA ATTRIBUTE command, or refer to the schema source file that contains the attribute definition, to determine whether the attribute allows multiple attribute values (multi-valued) or whether it just allows a single attribute value (single-valued). An attribute is multi-valued unless explicitly defined as being single-valued.

If the attribute is a single-valued attribute, re-execute the command specifying only one attribute value.

- A mandatory attribute is not specified. Using the DXIM Motif interface, the user has attempted to create a directory entry without providing values for all mandatory attributes.

  Check that the user has entered values in all the open text boxes.

DSA Error: Constraint Violation in Attribute Value *type = value*

Attribute Error: Constraint Violation

This may be returned under three circumstances:

- An attribute value does not meet the constraints imposed by the attribute definition in the directory schema For example, the attribute value is outside a range of values, or it may exceed the maximum size allowed.

  Check the value the user has entered for the specified attribute against the attribute definition, as described by the DXIM online help. The attribute definition defines whether any size restrictions apply to the attribute, or whether its value must be within a specified range of values.

  Correct the attribute as appropriate and retry the command.

- Multiple attribute values are being added to a single-valued attribute.

  If the user is adding multiple values, refer to the DXIM online help to determine if the attribute allows multiple attribute values (multi-valued) or whether it just allows a single attribute value (single-valued). By default, an attribute is multi-valued unless explicitly defined as being single-valued.

  If the attribute is a single-valued attribute, re-execute the command specifying only one attribute value.

- A mandatory attribute is not specified. Using the DXIM Motif interface, the user has attempted to create a directory entry without providing values for all mandatory attributes.

  Check that the user has entered values in all the open text boxes.

DSA Error: DIT Consistency Error; Directory Service Unable to Perform Operation

Service Error: DIT Consistency Error

An inconsistency has been detected in the DIT. This is possibly localized in an entry or set of entries. Check that the entries specified in the directory operation are correctly defined.

DSA Error: DSA is Unable to Proceed DSA Does Not Have Access to This Information

Service Error: Unable To Proceed

The DSA received a request which it cannot resolve directly. For example, it was asked to resolve a name within a naming context over which it has no administrative authority, or has followed a referral that does not lead to the requested entry being foun d.

DSA Error: Directory Service Busy, Please Try Later

Service Error: Busy

The Directory Service, or a part of it, is temporarily too busy to carry out the requested operation.

Allow time for some outstanding operations to complete, then retry the requested operation. If the request continually fails, contact VSI.

DSA Error: Directory Service Unable to Perform This Command Critical Extension Unavailable

Service Error: Unavailable Critical Extension This error is not returned by HP DSAs.

DSA Error: Directory Service Unavailable, Please Try Later

Service Error: Unavailable

There is a problem in the DSA which prevents it accepting the request from DXIM.

Check the DSA's event log for more specific information about DSA problems.

DSA Error: Directory Service is Unable to Perform this Command Invalid Reference Supplied

Service Error: Invalid Reference

The DSA was unable to perform the operation due to an invalid knowledge reference. Correct the relevant knowledge reference and retry the operation.

Check the DSA's event log for the Distributed Access Failure event. The event indicates the type of reference that is faulty (either Superior or Subordinate) and the name of that reference.

Check the knowledge reference for the following:

- Incorrect AE Title

  When you create a superior or subordinate reference, NCL checks your command to ensure it is syntactically correct, but it does not check to ensure that it is logically correct, that is, that the AE Title actually identifies a valid DSA. Therefore, an invalid knowledge reference is only detected when that reference is actually used.

  The Distributed Access Failure event indicates the Identifier (distinguished name) of the reference, and the AE Title and Presentation Address of the DSAs that hold the reference information. Use this information to check

that the AE Title of this reference identifies a valid DSA as follows. On the node where the event was issued, run NCL and enter one of the following commands, depending on the type of the faulty reference:

```
NCL> SHOW DSA SUPERIOR REFERENCE ALL ATTRIBUTES
NCL> SHOW DSA SUBORDINATE REFERENCE ALL ATTRIBUTES
```

Secondly, check the AE Title of the superior or subordinate DSA by starting NCL and entering the following command:

```
NCL> SHOW NODE node-name DSA AE TITLE
```

where *node-name* is the name of the node which contains the superior or subordinate DSA.

Ensure that each AE Title in the knowledge reference matches the AE Title of a subordinate or superior DSA.

- Incorrect Presentation Address

  Check that the presentation address you have specified in your superior or subordinate reference actually identifies a valid DSA, as follows.

  On the node where the event was issued, run NCL and enter one of the following commands depending on the type of the faulty reference:

  ```
  NCL> SHOW DSA SUPERIOR REFERENCE ALL ATTRIBUTES
  NCL> SHOW DSA SUBORDINATE REFERENCE ALL ATTRIBUTES
  ```

  Secondly, check the Presentation Address of the DSA holding the superior or subordinate entry, by starting NCL and entering the following command:

  ```
  NCL> SHOW NODE node-name DSA PRESENTATION ADDRESS
  ```

  where *node-name* is the name of the node which contains the DSA holding the superior or subordinate entry.

  Ensure that the Presentation Address of the knowledge reference matches the Presentation Address of the DSA. If the Presentation Address of the DSA has multiple NSAPs, the knowledge reference can contain a subset of the NSAPs. If the knowledge reference specifies NSAPs that are not present in the Presentation Address of the DSA or if the upper layer selectors do not match, then the knowledge reference is not correct and should be reconfigured.

- Logic error

  It is possible that both the AE Title and the Presentation Address are correct, but that the knowledge reference points to an entry that is not superior or subordinate to this one.

  Correct the knowledge reference by starting NCL, deleting the incorrect reference and replacing it with a correct reference.

DSA Error: Directory Service Unwilling to Perform Requested Operation

Service Error: Unwilling To Perform

This may be returned under two circumstances:

- A non-HP DSA is not willing to perform the operation because it requires excessive resources, or because it would violate administrative policy. This could involve another organization or an administrative authority.

  Contact the authority concerned and take action as appropriate. Otherwise, redirect the request to another DSA.

- The HP DSA containing the entry has one or more of its DSA authentication characteristic attributes set.

DSA Error: Entry Already Exists

Update Error: Entry Already Exists

The name specified already exists.

If the entry already exists, you cannot create another entry with the same distinguished name. Therefore, retry the command using a different name.

DSA Error: Entry Not Modified Modification Would Change Entry Name

Update Error: Not Allowed on RDN

The modification would alter an entry's relative distinguished name and is consequently not allowed.

Reissue the command without the attribute causing the error. If the user's intention was to change the entry's relative distinguished name, they must use the RENAME command.

DSA Error: Entry not Created or Modified Because of Naming Violation Entry Type Invalid at This Point in DIT

Update Error: Naming Violation

The request has been rejected as it would leave the DIT improperly structured. For example, adding an entry as a subordinate of an alias, or in a position in the DIT not permitted for a member of its object class.

Check the following:

- That the syntax of the command is correct

  Specifically, check the sequence of RDNs that form the distinguished name of the entry, to ensure they are correct and correspond with the structure rules defined in the directory schema. Structure rules define the relationships between different structural classes used within the Directory Service and therefore define an entry's position within the DIT. If the entry is not allowed in the proposed position, a naming violation error is returned.

  If necessary, correct the command line and retry the command.

  For more information on structure rules, refer to *VSI Enterprise Directory Management*.

- That the parent entry is not an alias

Enter the following command:

```
dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIASES
```

where `name` is the distinguished name of the parent of the entry the user tried to add. The distinguished name of the parent entry is the distinguished name of the entry the user tried to add, minus the last RDN.

If the entry is an alias entry, information about that alias is displayed, for example, the `Aliased Object Name` attribute.

If the parent entry is an alias entry, then you cannot create a subordinate entry of that alias, unless the DSA is configured using NCL to allow it. The Dereference Alias on Modify characteristic attribute controls whether the DSA permits alias names to be used in all modify commands, including CREATE commands.

- That the schema is compatible with the DSA schema, and is not corrupt

DSA Error: Entry Not Created or Modified Because of Object Class Violation Resultant Entry Would be Invalid for Entry Type

Update Error: Object Class Violation

This error may be returned under two circumstances:

- The user tried to create an entry that would be inconsistent with its object class definition.
- The user tried to modify an entry in such a way that it would leave the entry inconsistent with its object class definition.

Typical problems are:

- An attribute is not allowed for the specified object class
- A single-valued attribute has multiple values
- A mandatory attribute is missing

Use the DXIM SHOW SCHEMA CLASS command to display the schema

definition for an object of the class the user was trying to create. For example, if the user was trying to create an entry of class OrganizationalPerson, then you would enter the command:

```
dxim> SHOW SCHEMA CLASS ORGANIZATIONALPERSON
```

Ensure that:

- For create operations:
  - All attributes within the command are actually allowed for an entry of the specified object class.
  - A single-valued attribute is not being assigned multiple values.

    - Values for all mandatory attributes are specified in the command.
- For modify operations:
    - An attempt is not being made to delete a mandatory attribute.
    - Multiple attribute values are not being assigned to a single-valued attribute.
    - An attribute is not being added that is not allowed for an entry with the specified object class.
- For all operations, ensure that the schema used by DXIM is compatible with the DSA schema.

DSA Error: Internal Loop Detected, Directory Service Operation Failed

Service Error: Loop Detected

The DSA has detected that a loop exists within the directory or that the processing of this operation would result in a loop. A loop occurs when the processing of a directory request returns to a previously recorded state. It is caused by the incorrect use of aliases or knowledge information.

Follow the progress of the request through each RDN forming part of the entry's distinguished name. At each RDN, check for knowledge references or the use of alias entries, and, if foun d, make a note of their target entries. Check for aliases using the following DXIM command:

```
dxim> SHOW ENTRY name ALL ATTRIBUTES DONT DEREFERENCE ALIASES
```

where *name* is the name of the entry. If the entry is an alias, information about that alias is displayed, for example, the Aliased Object Name attribute, which contains the distinguished name of the target entry.

Through step-by-step tracking of the progress of the directory request, you should be able to detect the faulty alias or knowledge reference that caused the loop. Note, this may not necessarily be the distinguished name returned by the event.

Amend the alias or knowledge reference as appropriate.

If there are no faulty aliases and the knowledge references used to process the request are correct, contact VSI and report the error as a software problem.

DSA Error: Invalid Username or Password

Security Error: Invalid Credentials

The DSA has rejected an attempt to bind for one of the following reasons:

- The user specified a name and password which the DSA could not verify, or found to be incorrect.

  The DSA might have failed to find the entry that represents the user. This can happen if the entry does not exist, or the DSA fails to make a connection to another DSA that holds the entry.

  The DSA might have found the entry, but foun d that the password in the entry does not match the one specified by the user.

- The user did not identify themselves at all, or specified a name without a password, and the DSA has been configured to require all bind attempts to include names and passwords.

If the DSA has its Reader Names characteristic attribute set, then all binds must include a name and password. If the user does not supply both a name and password which the DSA can successfully verify, this error is returned.

- The user specified a valid name and password, but is not listed in the DSA Reader Names characteristic attribute.

  If that characteristic attribure is configured, binds are only possible for users who specify a name that is listed in the attribute or in the Writer Names attribute, and who specify a password that the DSA succeeds in matching against the named entry.

If the user specified the wrong name and password, they can try again. If the Reader Names attribute is configured, you can use NCL to add this user's name to either the Reader Names attribute or the Writer Names attribute, if that is appropriate for the user.

You might also consider removing the Reader Names and Writer Names attributes from the DSA, as they are a crude method of controlling access to a DSA. You should be able to provide more efficient and precise controls using the access control functionality described in *HP Enterprise Directory — Management*.

DSA Error: Matching Requested Not Appropriate for Attribute *attribute type*

Attribute Error: Inappropriate Matching

This may be returned under two circumstances:

- The type of matching specified in a DXIM SEARCH comman d does not apply for the attribute type.

  Four types of matching are available from HP DSAs: equality, ordering, substring, and approximate matching. However, not all attributes support all matching types. If you specify a matching type that is not supported for an attribute, this error is returned.

  Retry the command by specifying a different type of matching for that attribute or alternatively, retry the command using a different attribute on which to match.

  Refer to *VSI Enterprise Directory Management* for more information on attribute matching rules and search filters.

- The directory schema used by DXIM is not compatible with that being used by the DSA.

  Compare the schema text files being used by each system for differences. All schema files have the file extension ".sc" and are located in the directory pointed to by the logical name DXD $DIRECTORY.

  If the two schemas are different, amend the attribute definition in the schema being used by the client system, to correspond with the attribute definition being used by the server system. Do this by amending the appropriate schema file that contains the attribute definition and recompiling the schema. Refer to *VSI Enterprise Directory Management* for more information about amending schema files and compiling the schema.

  Note also, that if the schema being used by one client application is incompatible with the server's schema, there is the possibility that the schemas being used by all other client applications are also

incompatible with the server's schema. Therefore, it is worth checking each client schema against the server's schema to ensure compatibility.

DSA Error: Matching Requested Not Appropriate for Attribute Value *type = value*

Attribute Error: Inappropriate Matching

This may be returned under two circumstances:

- The matching rule specified in a DXIM SEARCH command does not apply for the attribute type.

  Four types of matching are available from HP DSAs: equality, ordering, substring, and approximate matching. However, not all attributes support all matching types. If you specify a matching type that is not supported for an attribute, this error is returned.

  Retry the command by specifying a different type of matching for that attribute or alternatively, retry the command using a different attribute on which to match.

  Refer to *VSI Enterprise Directory Management* for more information on attribute matching rules and search filters.

- The directory schema used by DXIM is not compatible with that being used by the DSA.

  Compare the schema text files being used by each system for differences. All schema files have the file extension ".sc" and are located in the directory pointed to by the logical name DXD $DIRECTORY.

  If the two schemas are different, amend the attribute definition in the schema being used by the client system, to correspond with the attribute definition being used by the server system. Do this by amending the appropriate schema file that contains the attribute definition and recompiling the schema. Refer to *VSI Enterprise Directory Management* for more information about amending schema files and compiling the schema.

  Note also, that if the schema being used by one client application is incom patible with the server's schema, there is the possibility that the schemas being used by all other client applications are also incompatible with the server's schema. Therefore, it is worth checking each client schema against the server's schema to ensure compatibility.

DSA Error: Modification Not Made because it Affects Several DSAs

Update Error: Request Affect s Multiple DSAs

The request would affect several DSAs and this is not allowed. For example, a Modify RDN request on an entry that happens to be a Naming Context name, would invalidate any references to that Naming Context and result in an error.

DSA Error: Modification Not Made; Object Class Attribute May Not be Changed

Update Error: Object Class Modification Prohibited

The `objectClass` attribute can only be modified to add or remove an auxiliary class from the entry. Other types of modification produce this error. For example, if you attempt to add or remove a structural class or alias class, the command fails. Structural and alias classes must be specified when an entry is created, and cannot be modified.

Change your command so that it does not attempt to modify the `objectClass` attribute, or so that the only change to that attribute involves an auxiliary class.

DSA Error: Modification Not Made; Request Only Valid for Entries with No Subordinates

Update Error: Not Allowed on Non Leaf

The user tried to delete or rename a directory entry that has subordinate entries and this is not allowed.

To continue with the operation, the user must first delete all the subordinate entries. You can view these subordinates by entering the following command:

```
dxim> SHOW SUBORDINATES name
```

where *name* is the distinguished name of the entry the user was attempting to delete or rename.

DSA Error: No Information Available Within Requested Scope

Service Error: Out of Scope

The directory cannot supply a Referral or Partial-Outcome Qualifier within the required scope (as defined by the Scope service control).

The Scope service control specifies whether a request must be satisfied within this DSA, within this management domain, within this country, or whether it can be distributed worldwide. All Referrals and Partial-Outcome Qualifiers returned by a DSA must be within the scope defined. Otherwise, this error is returned. By default, Scope is set so that a request can be distributed worldwide.

If the user is using the DXIM command line interface, display the current setting of the DXIM Scope service control as follows:

```
dxim> SHOW DEFAULTS
```

If necessary, amend the setting of the service control by entering one of the following commands:

```
dxim> SET DEFAULT DOMAIN SCOPE
dxim> SET DEFAULT COUNTRY SCOPE
dxim> SET DEFAULT WORLD SCOPE
```

If the user is using the DXIM windows interface, the Scope service control is specified in the `DUA.LocalScope` parameter of the system-wide DUA defaults file. Ensure that this parameter is set to FALSE, that is, the request can be distributed.

Note that HP's DSA does not honor the Scope service control, so never returns this error.

DSA Error: No Such Entry Exists Matched the Following Part of the Entry Name

Name Error: No Such Object

The Directory Service cannot find the specified entry. This may be due to the following reasons:

- No naming context has been created.

  You cannot create an entry until the naming context in which is resides has been created. Create the relevant Naming Context entity and then create the entry. See *VSI Enterprise Directory Management* for information about creating Naming Context entities.

- The entry has been deleted (Motif interface only).

  The information contained within the DXIM Browse window is correct at the time the information is extracted from the directory. However, information displayed in the window can become out of date if subsequent directory operations have been performed on any of the entries displayed.

  It may be that the entry the user has selected no longer exists. Check this by double clicking twice on the parent of the entry. Double click once to collapse the parent entry and once more to expand the parent entry. If the entry is no longer displayed, it indicates that the entry does not exist in the directory; no further action is necessary.

- The user has specified the root entry in a command where it is not appropriate.

  The root is not a real entry and can only be used for certain operations such as the SHOW SUBORDINATES command. If the root is specified as the target entry for other operations, you will receive this error message.

- The entry's distinguished name has been specified incorrectly.

  Check that the user has specified the entry's distinguished name correctly. You can check the existence of the entry by asking the directory service to display all the subordinates of the parent entry. Do this as follows:

  ```
  dxim> SHOW SUBORDINATES name
  ```

  where *name* is the distinguished name of the parent entry.

- The entry is a copy and the master entry has been deleted.

  Information displayed by both the DXIM command line interface and the DXIM windows interface may be from a shadow copy of a directory entry that has been created through replication. If the master copy of this directory entry has been deleted, then any modifications made to this entry will fail with this error message, since all modifications must be performed on the master entry and not the copy entry.

  No further action is necessary. You cannot update a copy entry. The copy entry will also be deleted when replication is next performed.

- The user does not have the necessary access control to allow them to display or manipulate the entry.

  Check that the user has the necessary access control rights to perform the operation.

DSA Error: Operation Requires Chaining

Service Error: Chaining Required

Chaining is essential to complete the operation but is prohibited by the Chaining service control.

Abandon the operation, or set the Chaining service control to allow chaining and retry the operation.

DSA Error: Operation Requires Authentication Please Supply Username and Password

Security Error: Inappropriate Authentication

The user has not supplied a username and password.

If they are using the DXIM command line interface, they must enter their username and password as part of the Bind request, for example:

```
dxim> BIND identifier TO ADDRESS address NAME username PASSWORD password
```

where *name* is the distinguished name by which the user wants to be authenticated to the DSA and *password* is the password associated with this distinguished name.

If they are using the DXIM windows interface, they must enter a valid username and password using the Authenticate... window.

DSA Error: Requested Operation Requires Signed Argument

Security Error: Protection Required

The Directory Service will not carry out the operation because the argument was not signed. This can only be returned when interworking with a non-HP DSA.

HP DSAs do not support digital signatures, therefore the information cannot be accessed.

DSA Error: Security Error, Request Not Carried Out

Security Error: No Information

The requested operation has produced a security error for which there is no information available.

Check that the user is supplying a username and password with the command. Retry the command. If the problem persists, contact VSI.

DSA Error: The Attribute *attribute type* Does Not Exist

Attribute Error: No Such Attribute or Value

This may be returned under three circumstances:

- One of the attributes specified as a command argument was not found in the specified entry. This error is only reported by the DSA if the command has an explicit list of attributes specified in the selection argument, and none of them is present in the entry.

  In most cases, this is not an error; the DSA is simply reporting the outcome of a request.

- The user does not have sufficient access rights to display or modify the specified attribute.

Check the command to ensure that the specified attribute is correct. If it is, display all the attributes of the specified entry using the following DXIM command:

```
dxim> SHOW ENTRY name ALL ATTRIBUTES MASTER INFORMATION
```

If the specified attribute is displayed as a result, it means that the user only has access rights to display the attribute and not modify it.

If the specified attribute is not displayed, it means that the attribute does not exist, or that the user has insufficient access rights to display or modify the attribute.

- The directory schema used by DXIM is not compatible with that being used by the DSA.

  Use the DXIM SHOW SCHEMA command on the system where you are running DXIM and on the DSA system, to compare the schema being used by both systems. Specifically, use the SHOW SCHEMA ALL ATTRIBUTES command to display the attribute definitions. Ensure that the definitions for the specified attribute are the same.

  Alternatively, compare the schema text files being used by each system for differences. All schema files have the file extension ".sc" and are located in the directory pointed to by the logical name DXD$DIRECTORY.

  If the two schemas are different, amend the attribute definition in the schema being used by the client system, to correspond with the attribute definition being used by the server system. Do this by amending the appropriate schema file that contains the attribute definition and recompiling the schema. Refer to *VSI Enterprise Directory Management* for more information about amending schema files and compiling the schema.

  Note also, that if the schema being used by one client application is incom patible with the server's schema, there is the possibility that the schemas being used by all other client applications are also incompatible with the server's schema. Therefore, it is worth checking each client schema against the server's schema to ensure compatibility.

DSA Error: The Attribute *attribute type* Does Not Have a Value of *attribute value*

Attribute Error: No Such Attribute or Value

This may be returned under the following circumstances:

- One of the attribute values specified in the command was not foun d in the specified entry.
- The user does not have sufficient access rights to display or modify the specified attribute value.

  Check the attribute values in the command line to ensure they are correct. If they are, display all the attributes of the specified entry using the following DXIM command:

  ```
  dxim> SHOW ENTRY name ALL ATTRIBUTES MASTER INFORMATION
  ```

  If the specified attribute value is displayed as a result, it means that the user only has access rights to display the attribute value and not modify it.

  If the specified attribute value is not displayed, it means that the attribute value does not exist, or that the user has insufficient access rights to display or modify the attribute value.

DSA Error: Time Limit Exceeded Before Operation Could be Completed

Service Error: Time Limit Exceeded

The operation could not be performed within the time limit specified by the DXIM Maximum Time service control, or the DUA.TimeLimit parameter specified in the DUA defaults file.

If the user is using the DXIM command line interface, display the current setting of the DXIM Maximum Time service control as follows:

```
dxim> SHOW DEFAULTS
```

Increase the value of the `Maximum Time` service control and retry the command. Alternatively, specify the `No Maximum Time` service control. This means that the directory service continues processing the request for as long as necessary.

If the user is using the DXIM windows interface, the `DUA.TimeLimit` parameter of the DUA defaults file determines how long is spent on processing the user's request. Increase the value of this parameter.

DSA Error: Username and Password Supplied Do Not Authorize a Suitable Level of Authentication for this Operation

Security Error: Inappropriate Authentication

The user supplied a username and password, but they were insufficient for the level of protection required to carry out the requested operation.

DSA Error: You Have Insufficient Access Rights for this Request

The user does not have sufficient access rights to carry out the operation. Refer to Section 6.1.1.

DXIM Cannot Access Browse Results

This error is caused by an internal error while processing the browse results. The results cannot be displayed. Contact VSI.

DXIM Cannot Convert the Name

This error indicates an error in services used by DXIM, or in allocating memory. Retry the command. If the error persists, make sure sufficient memory is available (see Chapter 7).

DXIM Converted the Name, but Data Was Lost

This error indicates a limitation within DECwindows Motif.

DXIM Does Not Support This Syntax for This Attribute

DXIM does not support this syntax.

DXIM Error: AFI Missing from NSAP Address in the Presentation Address

Check that the user has specified the Presentation Address correctly. See *VSI Enterprise Directory Management* for details.

DXIM Error: Cannot Initialize "current"

The DUA defaults file defines the domain root to be / (the global root). The file includes some space characters immediately after the / character. Edit the DUA defaults file to make sure that the / character is the last character on the line for the domain root default. Then invoke DXIM again.

DXIM Error: Cannot Open the Schema File *file*

DXIM cannot open DXD$SCHEMA.DAT.

Check that the schema binary file is in the correct directory and has the correct protection set (see Appendix A for details). Check that the schema binary file is not corrupt. See Section 2.7.3 for more information.

DXIM Error: Cannot Read the Schema File *file*

DXIM cannot read DXD$SCHEMA.DAT because it is in an invalid format. The software version of DXIM and the schema binary file do not match.

Make sure the schema source files are up to date and compatible with the schema used in other parts of the Directory Service. VSI recommends that you use the same schema throughout your directory. Recompile the schema using the schema compiler on the DXIM node.

DXIM Error: Command Exceeds Line Length Limit

A DXIM command line cannot exceed 1024 characters. To specify a command containing more than 1024 characters, use continuation marks (-) and continuation lines. No continuation line may contain more than 1024 characters. There is no limit to the number of continuation lines a command can contain.

DXIM Error: DXIM Internal Error

Contact VSI.

DXIM Error: IDI Missing from NSAP Address in the Presentation Address

Check the user has specified the Presentation Address correctly. See *VSI Enterprise Directory Management* for details.

DXIM Error: Insufficient Memory for DXIM to Proceed. Contact your System Manager

Refer to Section 7.5 for information on how to solve this problem.

DXIM Error: Insufficient Memory to Proceed With Request

Refer to Section 7.5 for information on how to solve this problem.

DXIM Error: Insufficient Memory. DXIM Cannot Proceed With This Request

Refer to Section 7.5 for information on how to solve this problem.

DXIM Error: Invalid Distinguished Name. No valid attribute values

The DUA defaults file specifies that the domain root is / (the global root). This error means that the / character is immediately followed by one or more space characters. Edit the DUA defaults file to make sure that the / character is the last character on the line that defines the domain root, and then invoke DXIM again.

DXIM Error: Invalid Session Information. Check the knownDSA Value in the DUA Defaults File

Check that the DUA.KnownDSA value is correctly defined in the DUA defaults file.

See *VSI Enterprise Directory Management* for details of the DUA defaults file.

DXIM Error: Specified Class Not Known to Schema

Check the schema used by DXIM is correct and is compatible with the DSA schema.

DXIM Error: The Data From the Directory Has an Invalid Encoding

Contact VSI.

DXIM Error: The Data From the Directory Has an Invalid Length Encoding

Contact VSI.

DXIM Error: This Command Cannot be Carried Out Because it Would Require DXIM to Open More Files than it is Allowed To

DXIM is temporarily unable to carry out the command. Wait until some outstanding operations have completed, and retry the command.

DXIM Error: You do Not Have a Default Name Set Up to Bind With

Modify the DUA defaults file to include a default bind name, or specify the name in the Bind command. See *VSI Enterprise Directory Management* for details of the DUA defaults file.

DXIM is Unable to Bind to the DSA

Refer to Section 2.7 for information about solving problems starting DXIM.

Library Error: Not Supported

Version 1.3 of the X.500 Directory Service application programming interface (XDS) supports the Cell Directory Service (CDS) as well as the VSI Enterprise Directory. This error indicates that an X.500 request has been directed to a CDS server, or that a CDS request has been directed to an X.500 DSA.

The most likely cause of this problem is that you are using an application that you have developed using the X.500 API. In this case, refer to *HP Enterprise Directory — Programming* and *HP Enterprise Directory — Programming Reference* for details of how the X.500 API supports both services. You need to ensure that requests are directed to the appropriate directory service. It is possible that your application is capable of binding to both services, and is currently bound to the wrong one.

This error should never appear in the DXIM command line interface, and is unlikely to appear in the DXIM windows interface. If it does appear, then you need to find out why the DXIM requests are being directed to CDS, and make sure that DXIM is bound to an X.500 DSA.

Maximum Number of Results for Operation Exceeded

Try again with a more specific search request.

No Match for Object Class Identifier

The object type of the entry you are using the DXIM Motif interface to create or modify is not in the local schema. Check the schema definition and update it if necessary.

No Name Form Match for the Entry

The DXIM Motif interface cannot find the Name Form of the entry in the schema, so cannot display the entry. Check the schema definition and update it if necessary.

Non-recoverable Internal Error Detected

Contact VSI.

Partial results displayed

Partial results can be displayed because the request has exceeded a service limit. There are three types of limit that can cause this message: time limits, size limits, and administration limits. The message indicates which of these limits was exceeded.

You might be able to get further results by repeating the request and specifying that there should be no limit. For example:

```
dxim> search /c=us/o=abacus where surname=Jones no maximum time
```

If the limit exceeded was an administration limit, then specifying world scope might provide further results.

If the partial results are accompanied by a continuation reference, see Section 5.2.

Referral

Referral Error

A Referral is returned when a request cannot be satisfied by this DSA and the DSA cannot chain the request. The message is accompanied by a continuation reference containing the Presentation Address and, optionally, the AE Title of one or more DSAs better able to perform the operation.

See Section 5.2 for details of how to deal with continuation references.

The Directory is Unable to Perform Required Operation

Service Error: Unwilling To Perform

This may be returned under the following circumstances:

- The user has access to both a VSI Enterprise Directory and a Cell Directory Service, and has directed an inappropriate request to one of those services.
- A non-HP DSA is not willing to perform the operation because it requires excessive resources, or because it would violate administrative policy. This could involve another organization or an administrative authority.

  Contact the manager of the other vendor's DSA to find out what limitations it has on the support of type of operation that failed.
- The DSA containing the entry has one or more of its DSA authentication characteristic attributes set.

  Refer to Section 6.5.1.

The Operation Reached the Directory Service Administrative Limits

The information you have requested is not available within your domain.

There is a Problem Starting DXIM

See Section 2.7 for details of how to solve problems starting DXIM.

Unable to Generate an OM Workspace

Contact VSI.

Unrecognized Directory Error

Contact VSI.

Chapter 10. Events and Counters

This chapter describes the events and counters issued by VSI Enterprise Directory. It describes how you can use these for supporting or solving problems with the system.

10.1. 10.1 Events

On OpenVMS systems, events are generated without manual intervention.

The table below lists the events generated by the Enterprise Directory.

Table 10.1. Enterprise Directory Events

Event	Refer to
Accounting Disabled	Section 10.1.1
Accounting Enabled	Section 10.1.2
Accounting File Rollover	Section 10.1.3
Accounting File Access Failure	Section 10.1.4
Accounting Records Discarded	Section 10.1.5
Authentication Failure	Section 10.1.6
Changes of State	Section 10.1.7
Communication Failure Event	Section 10.1.8
Create Failure	Section 10.1.9
Distributed Operation Failure	Section 10.1.10
Failure To Start Accounting Facility	Section 10.1.11
Internal Error	Section 10.1.12
Listen Failure	Section 10.1.13
Resource Exhausted	Section 10.1.14
Shadow Agreement Update Complete	Section 10.1.15
Shadow Agreement Update Failure	Section 10.1.16
Shadow Update Complete	Section 10.1.17
Shadow Update Failure	Section 10.1.18

10.1.1. Accounting Disabled

This event is generated when the DSA disables the accounting facility. The event contains the filename of the closed accounting file, that is, the accounting file that the DSA was using when the accounting facility was disabled.

10.1.2. Accounting Enabled

This event is generated when the DSA successfully enables the accounting facility and creates an accounting file. The event contains the filename of the created accounting file.

10.1.3. Accounting File Rollover

This event is generated when the DSA perform s either an unscheduled or scheduled accounting file rollover. Accounting file rollover is the process by which the DSA closes the current accounting file and creates a new accounting file. A scheduled accounting file rollover is one that is initiated by the setting of the Accounting Rollover Interval attribute. An unscheduled accounting file rollover is one that is initiated by the Accounting Rollover Unscheduled Time attribute.

The event contains the following information:

- The filename of the closed accounting file.
- The filename of the new accounting file.

10.1.4. Accounting File Access Failure

This event is generated when the DSA fails to access the Accounting file for any reason. The event contains the following information:

- What type of access the DSA was trying to make to the accounting file. This can be one of WRITE, OPEN, or CLOSE access.
- A system message that indicates why the DSA failed to access the accounting file. The system messages are described in the documentation for your operating system.
- The filename of the accounting file that the DSA failed to open.

When the DSA fails to access the Accounting file, it continues to process user requests but does not record any accounting information about those requests. Instead the DSA generates an Accounting Records Discarded event and, when it can access the accounting file again, creates a Discard record that indicates how many accounting records were discarded while the DSA could not access the accounting file.

10.1.5. Accounting Records Discarded

This event is generated when the DSA is forced to discard records without writing them to the accounting file. The reasons why the DSA may be forced to discard accounting files are described in Chapter 8. The event indicates the number of records discarded since the last record was successfully written to the accounting file.

10.1.6. Authentication Failure

This event is generated when the DSA fails to authenticate the originator of a Bind request. The event returns the following information:

- The reason why authentication failed. This may be one of the following:
    - Inaccessible Password

      The DSA could not verify the supplied password because the DSA containing the directory entry is not accessible. This might be a temporary problem, for example, the connection to that DSA has been lost, or it might be a more permanent problem, where the DSA holding the directory entry is not a trusted DSA.

- Incorrect Password

  The password supplied in the Bind request does not match the password stored in the directory entry identified by the supplied distinguished name.

- Password Verification Loop

  When a DSA attempts to verify a password, it might need to communicate with another DSA to access the directory entry that contains the password. This communication might also require the specification of a password, which must be verified by the second DSA. It is therefore possible that two DSAs will find themselves in a situation where each is waiting for the other to verify a password. If this happens, one of the DSAs detects the problem, and the authentication fails.

- Unknown User

  The distinguished name supplied in the Bind request does not identify an object within the directory.

- Unrecognized User

  The DSA does not recognize the distinguished name supplied in the Bind request, or the NSAP through which the Bind request was received.

- Information on the application or user that requested the operation. This comprises:
  - The application entity title of the DSA from which the Bind request was received.
  - The presentation address of the DSA from which the Bind request was received.
  - The distinguished name that was supplied in the Bind request, if any.
  - The directory protocol in use, that is DAP, DSP, DOP or DISP.

    DAP is the protocol used by directory applications to connect to a DSA. DSP is the protocol DSAs use to chain requests. DOP is the protocol DSAs use to manage shadowing agreements. DISP is the protocol DSAs use to replicate information.

Refer to Chapter 6 for information on how to recover from security problems.

10.1.7. Changes of State

This event is generated when the state of the DSA changes, either as a result of a management directive, or as a result of an operational problem. The event provides the following information:

- The old state of the DSA
- The new state of the DSA

No action is necessary.

10.1.8. Communication Failure Event

This event is generated for communication failure. The event has three arguments.

- Protocol

- CommsInterface
- Diagnostic

The Protocol is one of the following:

```
Directory Access Protocol
Directory System Protocol
DISP Protocol
DOP Protocol
LDAP Protocol
```

The CommsInterface informs you what network interface failed.

The Diagnostic text gives further information about the Communication Failure.

10.1.9. Create Failure

This event is generated when the DSA fails to create the DSA entity in response to the NCL CREATE DSA directive. The event indicates the reason why the DSA entity could not be created, and provides additional diagnostic information where possible.

The reason is one of the following:

- DIT Corrupt

  The DSA Information Tree is corrupt. Refer to Section 2.4.2.

- DIT Incompatible

  The DSA Information Tree is incompatible with this version of the DSA. Refer to Section 2.4.3.

- DSA Information Tree Schema Incompatible

  The DSA Information Tree contains information that is not defined in the schema. Refer to Section 2.4.6.

- License Check Failure

  The license check for the product has failed. Refer to Section 2.4.10.

- Schema Corrupt

  The directory schema is corrupt. Refer to Section 2.4.2.

- Schema Incompatible

  The directory schema is incompatible with this version of the DSA. Both the schema compiler and the DSA contain internal revision numbers that define the revision level of the software. Either the schema has been changed such that it is now incompatible with the database, or an old version of the schema compiler has been used to compile the schema files. See Section 2.4 for more information.

- System Error

  Contact VSI.

- Database Already in Use by Another DSA

Look in the database file DSA-information-tree.lock for the DSA that last locked the database.

10.1.10. Distributed Operation Failure

This event is generated when a distributed operation fails. The event message includes the following information:

- The reason for the failure. This is one of the following:
  - Communications Failure

    The DSA could not establish an association. The Communications Problem argument contains more information.
  - DSA Not Registered

    The DSA cannot find a directory entry for the DSA it is trying to communicate with, either because the entry does not exist or because access controls prevent the entry being seen.
  - Address Lookup Failure

    The DSA cannot obtain the presentation address of the remote DSA from the directory, because either the information does not exist or it is not accessible.
  - Authentication Failure

    The DSA could not establish an association with the target DSA due to an authentication failure. See Chapter 6 for information about security problems.
  - Invalid Reference

    The target DSA has returned an Invalid Reference error message.
  - DSA Not Trusted

    The target DSA does not trust this DSA, and considers that the binding requires trust.
  - A ROSE Reject status

    The target DSA has rejected the remote operation. This is normally due to a protocol error.
- If the problem was a Communications Problem, there is a diagnostic message and one of the following reasons:

  Fatal Interface Error
  Insufficient Resources
  Network Unavailable
  Address Already In Use Invalid AEI
  Transport Error
  System Error
  Invalid Transport Template
  Unknown Error
  ACSE User Reject

  See Chapter 3 for further details of these problems and how to solve them.

- The access point of the target DSA with which this DSA was attempting to communicate when the failure occurred.
- Information on the application or user that requested the operation. This comprises:
  - The application entity title of the DSA from which the Bind request was received.
  - The presentation address of the DSA from which the Bind request was received.
  - The distinguished name that was supplied in the Bind request, if any.
  - The directory protocol in use, that is, DAP, DSP, DOP, DISP, or the DEC Shadow Protocol.

10.1.11. Failure To Start Accounting Facility

This event is generated when the DSA fails to start the accounting facility. It indicates why the DSA failed to start the accounting facility, which can be for either one of the following two reasons:

- There was insufficient disk space available for the accounting facility. In this case, you need to release extra disk space on the disk that the accounting facility uses, and then start the accounting facility again. Moving or deleting accounting files might be the simplest solution.

  VSI recommends that you store accounting files on a different disk to the DSA files. To do so, define the logical DXD$ACCOUNTING to point to a directory on a different disk to the one that contains DXD$DIRECTORY.
- The DSA failed to create an accounting thread.

  Restart the accounting facility. If this problem happens frequently, report it to VSI.

10.1.12. Internal Error

This event is generated when the DSA detects an internal error.

Contact VSI.

10.1.13. Listen Failure

This event is generated when the DSA fails to set up its own presentation address for receiving communications. The event indicates the reason for the failure, and provides additional diagnostic information where possible. The reason is one of the following:

Fatal Interface Error
Insufficient Resources
Network Unavailable
Address Already In Use
Invalid AEI
Transport Error
System Error
Invalid Transport Template
Unknown Error
ACSE User Reject

See Chapter 3 for details of how to solve these problems.

10.1.14. Resource Exhausted

This event is generated when the DSA detects that a critical resource is temporarily exhausted. This may be due to one of the following reasons:

Insufficient Associations

Insufficient resources are available to process a Bind operation.

Insufficient Disk Space

Insufficient disk space remains to checkpoint the DSA's DIB fragment.

Insufficient License Capacity

The DSA is licensed for a limited number of entries, and the limit has been exceeded.

Insufficient Memory

Insufficient memory remains to process the operation.

Insufficient Threads

Insufficient processor threads remain to perform the requested operation.

Miscellaneous Resource Exhausted

Any other resource is exhausted.

Refer to Chapter 7 for information on how to solve problems associated with system resources.

10.1.15. Shadow Agreement Update Completed

This event is generated every time a DSA successfully creates, modifies, or deletes a shadowing agreement. A shadowing agreement describes how and when a DSA must replicate a given naming context to or from another DSA. The event provides the following information:

- The name of the naming context to which the agreement applies
- Whether this DSA is the supplier or consumer of the naming context
- The access point of the other DSA to which the agreement applies
- The identifier of the agreement that was successfully updated.

Each DSA manages its own shadowing agreements, and automatically communicates with other DSAs to coordinate agreements. This event is therefore an indication of normal operation.

No action is necessary.

10.1.16. Shadow Agreement Update Failure

This event is generated when a DSA fails to create, modify, or delete a shadowing agreement. When a DSA attempts to manage a shadowing agreement, it communicates with the other DSA to which the agreement applies. That DSA should also have generated this event.

The event provides the following information:

- One of the following problems:

Communications Problem
DOP error received
DOP error sent

- If the problem was a Communications Problem, then there is a diagnostic message, and one of the following reasons:

  Fatal Interface Error Insufficient Resources Network Unavailable
  Address Already In Use Invalid AEI
  Transport Error System Error
  Invalid Transport Template Unknown Error
  ACSE User Reject

  See Chapter 3 for further details of these problems and how to solve them.

- If the problem was either DOP error received or DOP error sent, one of the following reasons:

  Invalid ID

  A DSA has specified that it wants to modify an agreement that is not known to the other DSA. This problem is self correcting.

  Duplicate ID

  A DSA has attempted to establish a new agreement with an identifier that is already in use by the other DSA. This problem is self correcting.

  Unsupported Binding Type

  Another vendor's DSA is attempting to use DOP for a purpose other than the management of shadowing agreements. HP DSAs only support DOP for the management of shadowing agreements. Ask the manager of the other vendor's DSA to prevent this from recurring, if possible.

  Parameter Missing

  Some mandatory information is missing from the DOP operation. You should never see this error, as it indicates an incorrect use of the protocol. If it occurs, report the problem to VSI.

  Invalid Agreement

  This indicates that you have a DIT configuration problem. See Section 4.6.1 for details of how to solve this problem.

  Currently Not Decidable

  This error is generated if either DSA is unable to determine how to process the proposed DOP operation. An occasional occurrence of this problem requires no action. However, if the problem recurs frequently see Section 4.6.2.

- The name of the naming context to which the agreement applies
- Whether this DSA is consumer of supplier of the naming context
- The access point of the other DSA that the agreement applies to

- The identifier of the agreement that was not updated

10.1.17. Shadow Update Complete

This event is generated when the DSA has successfully updated a shadow naming context. Both the consumer DSA and the supplier DSA generate this event. The event provides some or all of the following information:

- The distinguished name of the naming context that was updated.
- An indication as to whether the DSA generating the event was the supplier DSA or the consumer DSA.
- The access point of the DSA with which the update was processed.
- The identifier of the agreement relating to the replication.
- The type of update that took place, which is one of:

  No changes

  This indicates that there have been no changes to the naming context since the previous update, so no replication takes place.

  Incremental

  This indicates that the changes to the naming context have been replicated to the consumer DSA.

  Total

  This indicates that the entire naming context has been replicated to the consumer. This may be necessary if this is the first replication attempt, or to recover from certain problem situations. For example, if the two DSAs disagree about when the last update occurred, a total update takes place.

No action is necessary.

10.1.18. Shadow Update Failure

This event is generated when the DSA fails to update a shadow naming context. Both the supplier and consumer DSAs can generate this event. The event provides some or all of the following information, depending on whether the DISP protocol or the DECshadow protocol was used:

- The reason why the update failed. This is one of the following:

  Cannot Read Supplier Address
  Communications Failure
  Consumer Not Authenticated DISP error received
  DISP error sent DIT incompatible
  Insufficient Resources Invalid Arguments
  Invalid Update Protocol Schema incompatible
  Supplier DSA Unavailable
  Supplier DSA rejected Connection Unexpected Failure
  Update Incompatible

- If the DISP protocol was used, the name of each naming context that failed to be updated.
- Whether the DSA generating the event was the supplier DSA or the consumer DSA.
- The access point of the DSA with which the update was processed.
- The identifier of the agreement for which replication failed.
- If the reason was DISP error received or DISP error sent, then one of the following reasons is specified:

  Full Update Required

  A full update will take place automatically. No action is required.

  Inactive Agreement

  A DSA has specified a shadowing agreement that is marked as inactive on the other DSA. This is a self correcting problem. The other DSA will shortly inform this DSA that the agreement is inactive. This might occur because the shadowing agreement is in the process of being deleted.

  Invalid Agreement ID

  A DSA has specified a shadowing agreement identifier that the other DSA does not recognize. This problem is self correcting.

  Invalid Information Received

  A DSA has received invalid information during replication. This should not happen between two HP DSAs, but might occur if interworking with another vendor's DSA. HP does not guarantee to be able to interwork with another vendor's implementation of DISP.

  Invalid Sequencing

  This problem is self correcting.

  Missed Previous

  The two DSAs disagree about when the last update occurred. This problem is usually self correcting. However, if it occurs frequently, use the UPDATE DSA to force a full update.

  Update Already Received

  If this problem occurs infrequently, it is self correcting. If it occurs frequently, use the UPDATE DSA command on the consumer DSA. This forces the two DSAs to reset the shadowing agreement, which should solve the problem.

  Unsuitable Timing

  The shadowing agreement is not yet in the correct state, so the supplier DSA is not ready to participate in replication. This is usually self correcting. However, if it occurs frequently, use the UPDATE DSA command to force a full update.

  Unsupported Strategy

  Another vendor's DSA has proposed a type of replication that is not supported by the HP DSA.

Unwilling To Perform

An unexpected condition has prevented replication. If this happens frequently, report it to VSI.

Refer to Section 4.5 for information on how to solve the problems that are not self correcting, or to Chapter 3 if the event states that a communications failure occurred.

10.2. Counters

Counters provide statistics concerning the performance and state of the VSI Enterprise Directory. Such statistics are useful when solving problems associated with the Enterprise Directory. Typical counters include the number of transactions that the Enterprise Directory has performed, the number of Bind failures that have occurred, or the number of referrals a DSA has generated.

There is also a counter associated with each event generated by the Enterprise Directory, which is incremented each time that event is generated.

You cannot change the values of the counters. However, you can display them using the SHOW directive of the NCL management utility. Counters are generated by the DSA and are set to zero when the DSA entity is enabled.

Counters are either status counters or error counters.

Status counters are those that are incremented when the Enterprise Directory perform s some routine task during its daily operation. Therefore, it is normal behavior for these counters to increase rapidly. Status counters do not indicate any problem with the Enterprise Directory and are generated to provide information.

Error counters are those that are incremented when the Enterprise Directory encounters an error, an unexpected event, or when it fails to process a given operation. A sudden change in the rate at which these counters are incremented might indicate a problem. Therefore, you need to understand that Error counters provide a means of assessing possible problems with the operation of the Enterprise Directory.

The Enterprise Directory counters are described in the table below. For each counter, there is a description of the counter, why it was generated, and what problems this may indicate. All counters are listed in alphabetical order in the table.

Table 10.2. Enterprise Directory Counters

Counter	Description
Abandon Failures	Error counter. The number of Abandon Failed Errors generated by the DSA since it was enabled. The Abandon service enables a user to indicate that they are no longer interested in the request that they sent to the DSA, perhaps because the request is taking too long.
Abandon Operations	Status counter. The number of directory ABANDON operations performed by the DSA since it was enabled. A directory ABANDON operation is used only to cancel a Read, Compare, List, or Search operation.
Accounting Disabled	Status counter. The number of Accounting Disabled events generated by the DSA.

Counter	Description
Accounting Enabled	Status counter. The number of Accounting Enabled events generated by the DSA.
Accounting File Access Failures	Error counter. The number of Accounting File Access Failure events generated by the DSA.
Accounting File Rollover	Status counter. The number of Accounting File Rollover events generated by the DSA. The DSA rolls over its accounting file periodically, according to the Accounting Rollover Interval attribute.
Accounting Records Discarded	Error counter. The number of accounting records discarded by the DSA. The DSA maintains a list of accounting records that are to be written to the accounting file. If the DSA cannot access the accounting file, then records are discarded if the list grows too long. Discarded records are not recoverable.
Accounting Start Failures	Error counter. The number of Accounting Start Failure events generated by the DSA. This indicates that the accounting facility failed to start when requested.
Add Entry Operations	Status counter. The number of directory add entry operations requested of the DSA since it was enabled. A directory add entry operation is used to add a new entry to the directory information base. For example, this counter increases with each DXIM CREATE ENTRY command. The counter does not distinguish between operations performed by this DSA, and operations that this DSA passed on to another DSA for processing.
Attribute Errors	Error counter. The number of Attribute Errors generated by the DSA since it was enabled. This counter increases when, for example, a user attempts to violate an attribute constraint, or show an attribute that does not exist.
Authentication Failures	Error counter. The number of Authentication Failure events generated by the DSA since it was enabled. An Authentication Failure event is issued when the DSA fails to authenticate the originator of a Bind request.
Chained Abandon Operations	Status counter. The number of CHAINED ABANDON operations requested of the DSA since it was enabled. A CHAINED ABANDON operation requests the DSA to cancel a CHAINED READ, CHAINED SEARCH, CHAINED COMPARE, or CHAINED LIST operation that was requested of it previously.
Chained Add Entry Operations	Status counter. The number of CHAINED ADD ENTRY operations requested of the DSA since it was enabled. A CHAINED ADD ENTRY

Counter	Description
	operation involves the local DSA receiving an ADD ENTRY operation from another DSA.
Chained Binds Accepted	Status counter. The number of Bind requests from other DSAs that have been accepted by the DSA. A chained bind enables communication between two DSAs.
Chained Binds Rejected	Error counter. The number of Bind requests from other DSAs that have been rejected by the DSA. A chained bind enables communication between two DSAs. A DSA can reject a chained bind if it is not ready for communication.
Chained Compare Operations	Status counter. The number of CHAINED COMPARE operations requested of the DSA since it was enabled. A CHAINED COMPARE operation involves the local DSA receiving a COMPARE operation from another DSA.
Chained List Operations	Status counter. The number of CHAINED LIST operations requested the DSA since it was enabled. A CHAINED LIST operation involves the local DSA receiving a LIST operation from another DSA.
Chained Modify Entry Operations	Status counter. The number of CHAINED MODIFY ENTRY operations requested of the DSA since it was enabled. A CHAINED MODIFY ENTRY operation involves the local DSA receiving a MODIFY ENTRY operation from another DSA.
Chained Modify RDN Operations	Status counter. The number of CHAINED MODIFY RDN operations requested of the DSA since it was enabled. A CHAINED MODIFY RDN operation involves the local DSA receiving a MODIFY RDN operation from another DSA.
Chained Operation Referrals	Status counters. The number of times the DSA has sent a referral to another DSA. The referral contains information about one or more DSAs that might have the information that satisfies a user request.
Chained Read Operations	Status counter. The number of CHAINED READ operations requested of the DSA since it was enabled. A CHAINED READ operation involves the local DSA receiving a READ operation from another DSA.
Chained Remove Entry Operations	Status counter. The number of CHAINED REMOVE ENTRY operations requested of the DSA since it was enabled. A CHAINED REMOVE ENTRY operation involves the local DSA receiving a REMOVE ENTRY operation, from another DSA.

Counter	Description
Chained Search Operations	Status counter. The number of CHAINED SEARCH operations requested of the DSA since it was enabled. A CHAINED SEARCH operation involves the local DSA receiving a SEARCH operation from another DSA.
Changes of State	Status counter. The number of Change of State events generated by the DSA since it was enabled. The event is generated when, for example, you use the ENABLE DSA directive. In this case, the DSA changes from state OFF to state ENABLING and then to state ON, causing two Change of State events.
Compare Operations	Status counter. The number of directory COMPARE operations performed by the DSA since it was enabled. A directory COMPARE operation is used to compare a given entry with a specified directory entry.
Create Failures	Error counter. The number of Create Failure events (see EVENTS) generated by the DSA. A Create Failure event is generated when the DSA fails to create the DSA entity. However, because a successful creation is required before you can see the counter, the value of this counter is always 0.
Creation Time	Status counter. The time and date at which the DSA was created.
DISP Binds Accepted	Status counter. The number of DISP binds the DSA has accepted. Each DISP bind enables the DSA to replicate information to or from another DSA. A single DISP bind might be used to replicate several naming contexts, and can be used bi-directionally.
DISP Binds Rejected	Error counter. The number of DISP binds the DSA has rejected. Each failed DISP bind was an attempt by another DSA to connect to this DSA for replication purposes. A bind might be rejected, for example, because the calling DSA failed to authenticate to this DSA.
Distributed Operation Failures	Error counter. The number of Distributed Operation Failure events generated by the DSA since it was enabled. A DSA generates a Distributed Operation Failure event when it fails to establish communication with another DSA for any reason. For example, network problems or authentication problems might cause the DSA to generate this event.
DOP Binds Accepted	Status counter. The number of DOP binds the DSA has accepted. Each DOP bind enables the DSA to coordinate shadowing agreements with the calling

Counter	Description
	DSA regarding the naming contexts that the two DSAs are configured to supply to each other. A single DOP bind might be used to manage more than one shadowing agreement, and can be used bi-directionally.
DOP Binds Rejected	Error counter. The number of DOP binds rejected by the DSA. Each failed DOP bind was an attempt by another DSA to connect to this DSA and coordinate a shadowing agreement. For example, the calling DSA might have wanted to ask this DSA to create a new shadowing agreement, or to amend an agreement to force replication of a naming context to happen before the scheduled time. A bind might be rejected, for example, because the calling DSA failed to authenticate itself to this DSA.
DUA Binds Accepted	Status counter. The number of Bind requests from directory applications that have been accepted by the DSA. For example, each successful DXIM BIND command causes this counter to increase by one.
DUA Binds Rejected	Error counter. The number of Bind requests from directory user applications (DUAs) that have been rejected by the DSA. For example, each unsuccessful DXIM BIND command causes this counter to increase by one.
Exhausted Resource	Error counter. The number of Resource Exhausted events generated by the DSA since it was enabled. A Resource Exhausted event is generated when a DSA detects that a critical resource is exhausted, preventing it from processing a requested operation.
Internal Errors	Error counter. The number of Internal Error events generated by the DSA since it was enabled. An Internal Error event is generated when the DSA detects an internal error.
LDAP Binds Accepted	Status counter. Records the number of Lightweight Directory Access Protocol (LDAP) Binds accepted by the DSA since it was enabled.
LDAP Binds Rejected	Status counter. Records the number of Lightweight Directory Access Protocol (LDAP) Binds rejected by the DSA since it was enabled.
List Operations	Status counter. The number of directory LIST operations performed by the DSA since it was enabled. A directory LIST operation is used to obtain a list of the immediate subordinates of a specified directory entry.

Counter	Description
Listen Failures	Error counter. The number of Listen Failure events generated by the DSA since it was enabled. The DSA generates this event when something prevents it from setting up its own presentation access point for receiving communications.
Modify Entry Operations	Status counter. The number of directory MODIFY ENTRY operations performed by the DSA since it was enabled. A directory MODIFY ENTRY operation is used to amend an existing directory entry. For example, this command increases with each DXIM MODIFY ENTRY and DXIM SET ENTRY command. The counter does not distinguish between operations performed by this DSA, and operations that this DSA passed on to another DSA for processing.
Modify RDN Operations	Status counter. The number of directory MODIFY RDN operations requested of the DSA since it was enabled. A directory MODIFY RDN operation is used to modify the relative distinguished name (RDN) of a directory entry. For example, this counter increases with each DXIM RENAME ENTRY command. The counter does not distinguish between operations performed by this DSA, and operations that this DSA passed on to another DSA for processing.
Name Errors	Error counter. The number of Name Errors generated by the DSA since it was enabled. This counter increases when, for example, a user specifies a name which is not the name of an entry.
Read Operations	Status counter. The number of directory READ operations performed by the DSA since it was enabled. A directory READ operation is used to extract information from a specified directory entry.
Referrals	Status counter. The number of Referrals generated by the DSA since it was enabled. A referral occurs when a DSA cannot satisfy an operation itself and therefore returns to the application, a reference to another DSA which it believes can process the operation.
Remove Entry Operations	Status counter. The number of directory REMOVE ENTRY operations requested of the DSA since it was enabled. A directory REMOVE ENTRY operation is used to remove a specified entry from the directory information base. For example, this counter increases with each DXIM DELETE ENTRY command. The counter does not distinguish between operations performed by this

Counter	Description
	DSA, and operations that this DSA passed on to another DSA for processing.
Results	Status counter. The number of results generated by the DSA since it was enabled. A result is the successful completion of a directory operation.
Search Operations	Status counter. The number of directory SEARCH operations performed by the DSA since it was enabled. A directory SEARCH operation is used to search a section of the directory information base for specific information.
Security Errors	Error counter. The number of security errors detected by the DSA since it was enabled. A security error occurs when, for example, an end user tries to perform an operation for which they are not authorized.
Service Errors	Error counter. The number of Service Errors generated by the DSA since it was enabled. A Service Error is issued when an error occurs related to the provision of a service.
Shadow Agreement Updates Completed	Status counter. The number of Shadow Agreement Update events generated by the DSA. The DSA generates the event when it succeeds in creating, modifying, or deleting a shadowing agreement. A shadowing agreement describes when and how a DSA must replicate a given naming context to or from another DSA.
Shadow Agreement Update Failures	Error Counter. The number of Shadow Agreement Update Failure events generated by the DSA. The event is generated when a DSA fails to create, modify, or delete a shadowing agreement.
Shadow Updates Completed	Status counter. The number of Shadow Update Complete events generated by the DSA since it was enabled. The DSA generates the Shadow Update Complete event when it succeeds in copying or updating a naming context from another DSA. If the DSA copies two or more naming contexts from another DSA, then the successful copying of each naming context causes an event.
Shadow Update Failures	Error counter. The number of Shadow Update Failure events generated by the DSA since it was enabled. The Shadow Update Failure event is generated when the DSA fails to update its copy of a naming context. If the DSA fails to copy two or more naming contexts from another DSA, then the failure to copy each naming context causes an event.
Update Errors	Error counter. The number of Update Errors generated by the DSA since it was enabled. This

Counter	Description
	counter increases when, for example, a user attempts to create an entry that already exists.

Appendix A. Enterprise Directory Files

This appendix shows the Enterprise Directory files on OpenVMS systems.

A.1. Files on an OpenVMS System

Table A.1 shows the Enterprise Directory files present on an OpenVMS system, and the location of these files. The files will not all necessarily be installed on your system. Which files are present depends on the components you have installed. Table A.2 shows which files are installed from which component saveset.

On a VAX cluster, all system area files are in the SYS$COMMON area. Previous versions of the Enterprise Directory installed some files in SYS$SPECIFIC. If you have upgraded from a previous version, you should delete any Enterprise Directory files (identifiable by the DXD$ prefix) from SYS $SPECIFIC.

Table A.1. File Locations on an OpenVMS System

File Name	Location	Description
COSINE.SC	DXD$DIRECTORY	Schema source file
DEC.SC	DXD$DIRECTORY	Schema source file
DIT.SC	DXD$DIRECTORY	Schema source file
DSA-INFORMATION-TREE.SNAPID[1]	DXD$DIRECTORY	File containing the most up-to-date snapshot identifier, *id*
DSA-IN FORMATION-TREE .SNAPSHOT*id*[1]	DXD$DIRECTORY	Checkpoint file of the DSA's DIB fragment
DSA-IN FORMATION-TREE .UPDATE*id*[1]	DXD$DIRECTORY	Journal file of all changes made to the DSA DIB fragment
DUA.SC	DXD$DIRECTORY	Schema source file
DXD$ACI_TEMPLATE.DXIM	DXD$DIRECTORY	ACI template file
DXD$COMMON_ SHUTDOWN.COM	SYS$STARTUP	Common shutdown file
DXD $COMMON_STARTUP.COM	SYS$STARTUP	Common startup file
DXD$DSA_S ERVER_MAIN_ V*.EXE	SYS$SYSTEM	DSA image
DXD$DSA_SHUTDOWN.COM	SYS$STARTUP	DSA shutdown file
DXD$DSA_SHUTDOWN.NCL	SYS$STARTUP	NCL commands to shutdown DSA
DXD$DSA_STARTUP.COM	SYS$STARTUP	DSA startup file
DXD$DSA_STARTUP.NCL	SYS$STARTUP	NCL commands to start up DSA
DXD $DSA_STARTUP_INPUT.COM	DXD$DIRECTORY	DSA startup file
DXD$DSA_STARTUP_ OUTPUT.LOG[1]	DXD$DIRECTORY	Log of DSA startups

File Name	Location	Description
DXD $DUA_CONFIGURE.COM	SYS$STARTUP	DUA configuration file
DXD$DUA_DEFAULTS.DAT[b]	DXD$DIRECTORY	System-wide DUA defaults
DXD$DXIM.UID	DECW$SYSTEM_DEFAULTS	UID file for the DXIM windows interface
DXD$DXIM_CLI.EXE[c]	SYS$SYSTEM	Image for the DXIM command line interface
DXD$DXIM_CLI.HLB	SYS$HELP	Help for DXIM command line interface utility
DXD$DXIM_MOTIF.EXE	SYS$SHARE	Image for the DXIM windows interface
DXD$DXIM_MOTIF.HLB	SYS$HELP	Help for DXIM windows utility
DXD$IVP.COM	SYS$TEST	IVP command file
DXD$IVP_API.DAT	SYS$SYSROOT: [SYSTEST.DXD]	Data file used by IVP
DXD$IVP_BASE.DAT	SYS$SYSROOT: [SYSTEST.DXD]	Data file used by IVP
DXD$IVP_DAF.DAT	SYS$SYSROOT: [SYSTEST.DXD]	Data file used by IVP
DXD$IVP_DSA.DAT	SYS$SYSROOT: [SYSTEST.DXD]	Data file used by IVP
DXD$IVP_LUC.DAT	SYS$SYSROOT: [SYSTEST.DXD]	Data file used by IVP
DXD$LLAPI_CMA_SHR.EXE	SYS$SHARE	CMA variant of the llapi shareable image
DXD$LLAPI_DEFINITIONS.H	SYS$LIBRARY	llapi include file
DXD$LLAPI_SHR.EXE	SYS$SHARE	Standard llapi shareable image
DXD $LOGICALS_STARTUP.COM	SYS$STARTUP	Command file to define logicals
DXD$LOOKUP_CLI.EXE	SYS$SYSTEM	Lookup Client command line image
DXD$LOOKUP_MOTIF.EXE	SYS$SYSTEM	Lookup Client graphical image
DXDLU.DECW$BOOK	SYS$HELP	Lookup Client graphical interface help
DXDLU.DEFAULTS	SYS$MANAGER	Lookup Client defaults file
DXDLU.HLB	SYS$HELP	Lookup Client command line help library
DXDLU.UID	DECW$SYSTEM_ DEFAULTS	Lookup Client graphical interface help
DXD$LUC_CONFIGURE.COM	SYS$STARTUP	Lookup Client configuration procedure
DXD$LUC_MSG.EXE	SYS$MESSAGE	Lookup Client message file

File Name	Location	Description
DXD$SCHEMA.DAT	DXD$DIRECTORY	Compiled Schema file
DXD$SCHEMA.SC	DXD$DIRECTORY	Top-level Schema source file
DXD $SCHEMA_COMPILER.EXE	SYS$SYSTEM	Schema compiler
DXD$SERVER_LOGIN.COM	SYS$SYSTEM	Login.com for DXD$SERVER account
DXD $TEMPLATE_STARTUP.NCL	SYS$STARTUP	NCL commands to create OSI transport templates
DXD$XDS_SHR.EXE	SYS$SHARE	XDS shareable image
ENTRUST.SC	DXD$DIRECTORY	Schema source file
MTS.SC	DXD$DIRECTORY	Schema source file
QUIPU.SC	DXD$DIRECTORY	Schema source file
X400.SC	DXD$DIRECTORY	Schema source file
X500.SC	DXD$DIRECTORY	Schema source file
XDS.H	SYS$LIBRARY	XDS header file containing interface functions, structures and defined constants
XDSBDCP.H	SYS$LIBRARY	XDS header file containing object identifiers and other values for the Basic Directory Contents Package
XDSDEC.H	SYS$LIBRARY	XDS header file containing support for HP extensions to XDS
XDSMDUP.H	SYS$LIBRARY	XDS header file containing object identifiers and other values for the Message Handling System Directory User Package
XDSSAP.H	SYS$LIBRARY	XDS header file containing object identifiers and other values for the Strong Authentication Package
XOM.H	SYS$LIBRARY	Object management header file containing structures and defined constants
XOMI.H	SYS$LIBRARY	Object management header file containing interface functions and internal structures

[1]Generated the first time the DSA entity is created

[b]Generated by DXD$DUA_CONFIGURE.COM

[c]This executable is installed with PRMMBX and SYSLCK privileges.

Table A.2. Files Installed from Each Saveset on an OpenVMS System

File Name	BASE	SRVR	DXIM	API	DXDLU
COSINE.SC	Y	Y	Y	-	-
DEC.SC	Y	Y	Y	-	-
DIT.SC	Y	Y	Y	-	-
DUA.SC	Y	Y	Y	-	-
DXD$ACI_TEMPLATE.DXIM	Y	Y	Y	-	-
DXD$COMMON_STARTUP.COM	Y	Y	Y	-	-
DXD$COMMON_SHUTDOWN.COM	Y	Y	Y	-	-
DXD$DSA_S ERVER_MAIN.EXE	-	Y	-	-	-
DXD$DSA_S HUTDOWN.COM	-	Y	-	-	-
DXD$DSA_S HUTDOWN.NCL	-	Y	-	-	-
DXD$DSA_STARTUP.COM	-	Y	-	-	-
DXD$DSA_STARTUP.NCL	-	Y	-	-	-
DXD$DSA_STARTUP_IN PUT.COM	-	Y	-	-	-
DXD$DUA_CONFIGURE.COM	Y	Y	Y	-	-
DXD$DUA_DEFAULTS.DAT	Y	Y	Y	-	-
DXD$DXIM.UID	-	-	Y	-	-
DXD$DXIM_CLI.EXE	-	-	Y	-	-
DXD$DXIM_CLI.HLB	-	-	Y	-	-
DXD$DXIM_MOTIF.EXE	-	-	Y	-	-
DXD$DXIM_MOTIF.HLB	-	-	Y	-	-
DXD$IVP.COM	Y	-	-	-	-
DXD$IVP_API.DAT	-	-	-	Y	-
DXD$IVP_BASE.DAT	Y	-	-	-	-
DXD$IVP_DAF.DAT	-	-	Y	-	-
DXD$IVP_DSA.DAT	-	Y	-	-	-
DXD$IVP_LUC.DAT	-	-	-	-	Y
DXD$LLAPI_CMA_S HR.EXE	Y	-	-	-	-
DXD$LLAPI_DEFINITIONS.H	Y	Y	Y	-	-
DXD$LLAPI_SHR.EXE	Y	Y	Y	-	-
DXD$LOGICALS_STARTUP.COM	Y	Y	Y	-	-
DXD$LOOKUP_CLI.EXE	-	-	-	-	Y
DXD$LOOKUP_MOTIF.EXE	-	-	-	-	Y
DXDLU.DECW$BOOK	-	-	-	-	Y
DXDLU.DEFAULTS	-	-	-	-	Y
DXDLU.HLB	-	-	-	-	Y
DXDLU.UID	-	-	-	-	Y
DXD$LUC_CONFIGURE.COM	-	-	-	-	Y

File Name	BASE	SRVR	DXIM	API	DXDLU
DXD$LUC_MSG.EXE	-	-	-	-	Y
DXD$SCHEMA.DAT	Y	Y	Y	-	-
DXD$SCHEMA.SC	Y	Y	Y	-	-
DXD$SCHEMA_COMP ILER.EXE	Y	Y	Y	-	-
DXD$SERVER_LOGIN.COM	-	Y	-	-	-
DXD$TEMPLATE_STARTUP.NCL	Y	Y	Y	-	-
DXD$XDS_SHR.EXE	Y	Y	Y	-	-
ENTRUST.SC	Y	Y	Y	-	-
MTS.SC	Y	Y	Y	-	-
QUIPU.SC	Y	Y	Y	-	-
X400.SC	Y	Y	Y	-	-
X500.SC	Y	Y	Y	-	-
XDS.H	-	-	-	Y	-
XDSBDCP.H	-	-	-	Y	-
XDSDEC.H	-	-	-	Y	-
XDSMDUP.H	-	-	-	Y	-
XDSSAP.H	-	-	-	Y	-
XOM.H	-	-	-	Y	-
XOMI.H	-	-	-	Y	-

Table A.3 shows the Enterprise Directory files used on an OpenVMS system, the location of these files on disk, and the protection applied to each file.

Table A.3. File Protections on an OpenVMS System

File Name	Protection(System, Owner, Group, World)
COSINE.SC	RWED, RWED, RWED, RE
DEC.SC	RWED, RWED, RWED, RE
DIT.SC	RWED, RWED, RWED, RE
DSA-IN FORMATION-TREE.snapid	RWED, RWED,,
DSA-IN FORMATION-TREE.snapshot *id*	RWED, RWED,,
DSA-IN FORMATION-TREE.update *id*	RWED, RWED,,
DUA.SC	RWED, RWED, RWED, RE
DXD$ACI_TEMPLATE_DXIM	RWED, RWED, RWED, RE
DXD$COMMON_SHUTDOWN.COM	RWED, RWED, RWED,
DXD$COMMON_STARTUP.COM	RWED, RWED, RWED,
DXD$DSA_SERVER_MAIN.EXE	RWED, RWED, RWED, RE
DXD$DSA_SHUTDOWN.COM	RWED, RWED, RWED,
DXD$DSA_SHUTDOWN.NCL	RWED, RWED, RWED,
DXD$DSA_STARTUP.COM	RWED, RWED, RWED,

File Name	Protection(System, Owner, Group, World)
DXD$DSA_STARTUP.NCL	RWED, RWED, RWED,
DXD$DSA_STARTUP_INPUT.COM	RWED, RWED, RWED, RE
DXD$DSA_STARTUP_OUTPUT.LOG	RWED, RWED, RE,
DXD$DUA_CONFIGURE.COM	RWED, RWED, RWED,
DXD$DUA_DEFAULTS.DAT	RWED, RWED, RWED, R
DXD$DXIM.UID	RWED, RWED, RWED, RE
DXD$DXIM_CLI.EXE	RWED, RWED, RWED, RE
DXD$DXIM_CLI.HLB	RWED, RWED, RWED, RE
DXD$DXIM_MOTIF.EXE	RWED, RWED, RWED, RE
DXD$DXIM_MOTIF.HLB	RWED, RWED, RWED, RE
DXD$IVP.COM	RWED, RWED, RWED,
DXD$IVP_API.DAT	RWED, RWED, RWED,
DXD$IVP_BASE.DAT	RWED, RWED, RWED,
DXD$IVP_DAF.DAT	RWED, RWED, RWED,
DXD$IVP_DSA.DAT	RWED, RWED, RWED,
DXD$IVP_LUC.DAT	RWED, RWED, RWED,
DXD$LLAPI_CMA_SHR.EXE	RWED, RWED, RWED,RE
DXD$LLAPI_DEFINITIONS.H	RWED, RWED, RWED, RE
DXD$LLAPI_SHR.EXE	RWED, RWED, RWED, RE
DXD$LOGICALS_STARTUP.COM	RWED, RWED, RWED,
DXD$LOOKUP_CLI.EXE	RWED, RWED, RWED, RE
DXD$LOOKUP_MOTIF.EXE	RWED, RWED, RWED, RE
DXDLU.DECW$BOOK	RWED, RWED, RWED, RE
DXDLU.DEFAULTS	RWED, RWED, RWED, R
DXDLU.HLB	RWED, RWED, RWED, RE
DXDLU.UID	RWED, RWED, RWED, RE
DXD$LUC_CONFIGURE.COM	RWED, RWED, RWED,
DXD$LUC_MSG.EXE	RWED, RWED, RWED, RE
DXD$SCHEMA.DAT	RWD, RWD, R, R
DXD$SCHEMA.SC	RWED, RWED, RWED, RE
DXD$SCHEMA_COMPILER.EXE	RWED, RWED, RWED, RE
DXD$SERVER_LOGIN.COM	RWED, RWED, RWED, RE
DXD$TEMPLATE_STARTUP.NCL	RWED, RWED, RWED,
DXD$XDS_SHR.EXE	RWED, RWED, RWED, RE
ENTRUST.SC	RWED, RWED, RWED, RE
MTS.SC	RWED, RWED, RWED, RE
QUIPU.SC	RWED, RWED, RWED, RE
X400.SC	RWED, RWED, RWED, RE

File Name	Protection(System, Owner, Group, World)
X500.SC	RWED, RWED, RWED, RE
XDS.H	RWED, RWED, RWED, RE
XDSBDCP.H	RWED, RWED, RWED, RE
XDSDEC.H	RWED, RWED, RWED, RE
XDSMDUP.H	RWED, RWED, RWED, RE
XDSSAP.H	RWED, RWED, RWED, RE
XOM.H	RWED, RWED, RWED, RE
XOMI.H	RWED, RWED, RWED, RE

Appendix B. Summary of Enterprise Directory NCL Directives

This appendix provides you with a summary of the NCL directives and attributes associated with each entity or subentity of VSI Enterprise Directory.

- Section B.1 gives an overview of the X.500 NCL directives for a DSA entity.
- Section B.2 gives an overview of the X.500 NCL directives for a Superior Reference subentity.
- Section B.3 gives an overview of the X.500 NCL directives for a Subordinate Reference subentity.
- Section B.4 gives an overview of the X.500 NCL directives for a Naming Context subentity.
- Section B.5 gives an overview of the X.500 NCL directives for an Accessor subentity.

For a more complete description of these NCL directives, including the full syntax and examples, see the Directory Module section of the online help for NCL.

B.1. NCL Directives for the DSA Entity

Table B.1 lists the X.500 NCL directives for a DSA entity.

Table B.1. NCL Directives for the DSA Entity

Directive	Description
ADD DSA characteristic	Add a new value to a characteristic attribute.
CREATE DSA	Create a DSA entity.
CREATE DSA FROM SNAPSHOT	Create a DSA using a snapshot file.[1]
DELETE DSA	Delete a DSA.
DELETE DSA TO SNAPSHOT	Delete a DSA and write a snapshot file.[1]
DISABLE DSA	Disable a DSA entity for communication.
ENABLE DSA	Enable a DSA entity for communication.
REMOVE DSA characteristic	Remove a value from a characteristic attribute.
SET DSA characteristic	Set a specific DSA characteristic attribute.
SHOW DSA	Display all information about a DSA.
SHOW DSA attribute	Displays a specific DSA attribute.
SHOW DSA ALL ATTRIBUTES	Displays all DSA attributes.
SHOW DSA ALL CHARACTERISTICS	Displays all DSA characteristic attributes. SHOW DSA ALL STATUS Displays all DSA status attributes.
SHOW DSA ALL COUNTERS	Displays all DSA counters.
UPDATE DSA supplier	Causes an unscheduled update between the target DSA and the DSA identified in the supplier argument.[2]

[1]On OpenVMS, DSAs use snapshot files only.

[2]Replication is automated in this version. The UPDATE DSA directive should only be required when you first implement replication, or for some problem solving situations.

Table B.2 lists the DSA attributes, showing which directives can be used with each attribute.

Table B.2. DSA Entity Attributes

Attribute	Set	Show	Add	Remove	Attribute Type
Abandon Failures	-	Y	-	-	counter
Abandon Operations	-	Y	-	-	counter
Accounting Facility	Y	Y	-	-	characteristic
Accounting Rollover Interval	Y	Y	-	-	characteristic
Accounting Options	Y	Y	Y	Y	characteristic
Add Entry Operations	-	Y	-	-	counter
AE Title	Y	Y	-	-	characteristic
Attribute Count	-	Y	-	-	status
Attribute Errors	-	Y	-	-	counter
Authentication Failures	-	Y	-	-	counter
Chained Abandon Operations	-	Y	-	-	counter
Chained Add Entry Operations	-	Y	-	-	counter
Chained Binds Accepted	-	Y	-	-	counter
Chained Binds Rejected	-	Y	-	-	counter
Chained Compare Operations	-	Y	-	-	counter
Chained List Operations	-	Y	-	-	counter
Chained Modify Entry Operations	-	Y	-	-	counter
Chained Modify RDN Operations	-	Y	-	-	counter
Chained Operation Referrals	-	Y	-	-	counter
Chained Read Operations	-	Y	-	-	counter
Chained Remove Entry Operations	-	Y	-	-	counter
Chained Search Operations	-	Y	-	-	counter
Changes of State	-	Y	-	-	counter
Communication Failures	-	Y	-	-	counter
Compare Operations	-	Y	-	-	counter
Create Failures	-	Y	-	-	counter
Creation Time	-	Y	-	-	counter
Dereference Aliases on Modify	Y	Y	-	-	characteristic
Distributed Operations Failures	-	Y	-	-	counter
DIT Check Interval	Y	Y	-	-	characteristic
DIT Check Last Time	-	Y	-	-	characteristic
DIT Check Unscheduled Time	Y	Y	-	-	characteristic
DIT Check Start Time	Y	Y	-	-	characteristic
DIT Check Window	Y	Y	-	-	characteristic
DIT Memory Occupancy	-	Y	-	-	status
DISP Binds Accepted	-	Y	-	-	counter

Attribute	Set	Show	Add	Remove	Attribute Type
DISP Binds Rejected	-	Y	-	-	counter
DOP Binds Accepted	-	Y	-	-	counter
DOP Binds Rejected	-	Y	-	-	counter
DUA Binds Accepted	-	Y	-	-	counter
DUA Binds Rejected	-	Y	-	-	counter
Entry Count	-	Y	-	-	status
Exhausted Resource	-	Y	-	-	counter
Idle Disconnect Timer	Y	Y	-	-	characteristic
Internal Errors	-	Y	-	-	counter
LDAP Binds Accepted	-	Y	-	-	counter
LDAP Binds Rejected	-	Y	-	-	counter
LDAP Port	Y	Y	-	-	characteristic
LDAP Idle Disconnect Timer	Y	Y	-	-	characteristic
List Operations	-	Y	-	-	counter
Listen Failures	-	Y	-	-	counter
Modify Entry Operations	-	Y	-	-	counter
Modify RDN Operations	-	Y	-	-	counter
Name Errors	-	Y	-	-	counter
Password	Y	-	-	-	characteristic
Presentation Address	Y	Y	-	-	characteristic
Prohibit Chaining	Y	Y	-	-	characteristic
Reader Names	Y	Y	Y	Y	characteristic
Reader NSAPs	Y	Y	Y	Y	characteristic
Read Only DSA Names	Y	Y	Y	Y	characteristic
Read Only DSA NSAPs	Y	Y	Y	Y	characteristic
Read Operations	-	Y	-	-	counter
Referrals	-	Y	-	-	counter
Remove Entry Operations	-	Y	-	-	counter
Results	-	Y	-	-	counter
Schema Check on Modify	Y	Y	-	-	characteristic
Search Operations	-	Y	-	-	counter
Security Errors	-	Y	-	-	counter
Service Errors	-	Y	-	-	counter
Shadow Agreement Update Failures	-	Y	-	-	counter
Shadow Update Failures	-	Y	-	-	counter
Shadow Agreement Updates Completed	-	Y	-	-	counter
Shadow Updates Completed	-	Y	-	-	counter
Size Limit	Y	Y	-	-	characteristic

Attribute	Set	Show	Add	Remove	Attribute Type
State	-	Y	-	-	status
Time Limit	Y	Y	-	-	characteristic
Trusted DSA Names	Y	Y	Y	Y	characteristic
Trusted DSA NSAPs	Y	Y	Y	Y	characteristic
UID	-	Y	-	-	status
Unique Value Count	-	Y	-	-	status
Update Errors	-	Y	-	-	counter
Version	-	Y	-	-	characteristic
Volatile Modifications	Y	Y	-	-	characteristic
Writer Names	Y	Y	Y	Y	characteristic
Writer NSAPs	Y	Y	Y	Y	characteristic

B.2. NCL Directives for the Superior Reference Subentity

Table B.3 lists the X.500 NCL directives for a Superior Reference subentity. The Superior Reference subentity has one characteristic attribute, Access Point.

Table B.3. NCL Directives for Superior Reference

Directive	Description
ADD DSA SUPERIOR REFERENCE ACCESS POINT access-point	Add a new value to the characteristic attribute.
CREATE DSA SUPERIOR REFERENCE ACCESS POINT access-point	Create a Superior Reference.
DELETE DSA SUPERIOR REFERENCE ACCESS POINT access-point	Delete a Superior Reference.
REMOVE DSA SUPERIOR REFERENCE ACCESS POINT access-point	Delete a Superior Reference.
SHOW DSA SUPERIOR REFERENCE	Display all information about a Superior Reference.
SET DSA SUPERIOR REFERENCE ACCESS POINT access-point	Set the Access Point characteristic attribute.

B.3. NCL Directives for the Subordinate Reference Subentity

Table B.4 lists the X.500 NCL directives for a Subordinate Reference subentity.

Table B.4. NCL Directives for Subordinate Reference Entity

Directive	Description
ADD DSA SUBORDINATE REFERENCE characteristic	Add a new value to a characteristic attribute.

Directive	Description
CREATE DSA SUBORDINATE REFERENCE characteristic	Create a Subordinate Reference.
DELETE DSA SUBORDINATE REFERENCE characteristic	Delete a Subordinate Reference.
REMOVE DSA SUBORDINATEREFERENCE characteristic	Remove a value from a characteristic attribute.
SHOW DSA SUBORDINATE REFERENCE	Display all information about a Subordinate Reference.
SHOW DSA SUBORDINATE REFERENCE characteristic	Display a specific Subordinate Reference characteristic attribute.
SHOW DSA SUBORDINATE REFERENCEALL CHARACTERISTICS	Display all Subordinate Reference characteristic attributes
SET DSA SUBORDINATE REFERENCE characteristic	Set a specific Subordinate Reference characteristic attribute.

Table B.5 lists the Subordinate Reference characteristic attributes, showing which directives can be used with which attribute.

Table B.5. Subordinate Reference Entity Characteristic Attributes

Attribute	Set	Show	Add	Remove	Attribute Type
Access Point	Y	Y	Y	Y	characteristic
Copy Access Point	Y	Y	Y	Y	characteristic

B.4. NCL Directives for the Naming Context Subentity

Table B.6 lists the X.500 NCL directives for a Naming Context subentity.

Table B.6. NCL Directives Naming Context Entity

Directive	Description
ADD DSA NAMING CONTEXT characteristic	Add a new characteristic attribute value.
CREATE DSA NAMING CONTEXT dist-name	Create a Naming Context with the specified distinguished name.
CREATE DSA NAMING CONTEXT dist-name characteristic	Create a Naming Context with the specified distinguished name and characteristic.
DELETE DSA NAMING CONTEXT dist-name	Delete a Naming Context.
REMOVE DSA NAMING CONTEXT dist-name characteristic	Remove a value from a characteristic attribute.
SHOW DSA NAMING CONTEXT dist-name	Display all information about a Naming Context.
SHOW DSA NAMING CONTEXT dist-name characteristic	Display all information about the specified attribute of the Naming Context.
SHOW DSA NAMING CONTEXT ALL CHARACTERISTICS	Display all the Naming Context characteristic attributes.

Directive	Description
SET DSA NAMING CONTEXT dist-name characteristic	Set a specific Naming Context characteristic attribute.

Table B.7 lists the Naming Context characteristic attributes, showing which directives can be used with which attribute.

Table B.7. Naming Context Entity Characteristic Attributes

Attribute	Set	Show	Add	Remove	Attribute Type
Consumer Access Point	Y	Y	Y	Y	characteristic
Master Access Point	-	Y	-	-	characteristic
Supplier Access Point	-	Y	-	-	characteristic

B.5. NCL Directives for the Accessor Subentity

Table B.8 lists the X.500 NCL directives for an Accessor subentity. The Accessor subentity has only one characteristic attribute, Password.

Table B.8. NCL Directives for Accessor Entity

Directive	Description
CREATE DSA ACCESSOR name PASSWORD password	Create an Accessor subentity with the specifieddistinguished name and password
DELETE DSA ACCESSOR name	Delete an Accessor subentity.
SHOW DSA ACCESSOR name	Display information about the Accessor subentity.
SET DSA ACCESSOR name PASSWORD password	Assign a value to the Password characteristic attribute.

Note that the Accessor entity is volatile information. If you stop and restart the DSA, the Accessor entity is deleted.