Sun Java™ System

Directory Server 5.2
Installation and Migration Guide

2005Q1

Contents

List of Tables

List of Figures

Preface

This guide contains information about how to upgrade Directory Server installations, how to migrate data from an existing Directory Server instance to an current instance, and how to use Directory Server in a Sun Cluster environment.

For information about how to access Sun™ documentation and how to use Sun documentation, see the following sections:

- Conventions
- Related Books
- Documentation, Support, and Training
- Related Third-Party Web Site References
- Sun Welcomes Your Comments

Conventions

Table 1 describes the typeface conventions used in this document.

Table 1   Typeface Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, on-screen computer output, sample code.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
AaBbCc123 (Monospace bold)	What you type, as contrasted with on-screen computer output.	`% `**`su`** `Password:`

Table 1 Typeface Conventions *(Continued)*

Typeface	Meaning	Examples
AaBbCc123 (Italic)	Book titles.	Read Chapter 6 in the *Developer's Guide*.
	New words or terms.	These are called *class* options.
	Words to be emphasized.	You *must* be superuser to do this.
	Command-line variables to be replaced by real names or values.	The file is located in the *ServerRoot* directory.

Table 2 describes placeholder conventions used in this guide.

Table 2 Placeholder Conventions

Item	Meaning	Examples
install-dir	Placeholder for the directory prefix under which software binaries reside after installation.	The default *install-dir* prefix on Solaris systems is `/`. The default *install-dir* prefix on Red Hat systems is `/opt/sun`.
ServerRoot	Placeholder for the directory where server instances and data reside. You can manage each server under a *ServerRoot* remotely through your client-side Server Console. The Server Console uses the server-side Administration Server to perform tasks that must execute directly on the server-side system.	The default *ServerRoot* directory is `/var/opt/sun/serverroot`.
`slapd-`*serverID*	Placeholder for the directory where a specific server instance resides under the *ServerRoot* and its associated data resides by default.	The default *serverID* is the host name.

Table 3 describes the symbol conventions used in this book.

Table 3 Symbol Conventions

Symbol	Meaning	Notation	Example
[ ]	Contain optional command options.	O[*n*]	O4, O
{ }	Contain a set of choices for a required command option.	d{y|n}	dy
|	Separates command option choices.		

Table 3   Symbol Conventions *(Continued)*

Symbol	Meaning	Notation	Example
+	Joins simultaneous keystrokes in keyboard shortcuts that are used in a graphical user interface.		Ctrl+A
-	Joins consecutive keystrokes in keyboard shortcuts that are used in a graphical user interface.		Esc-S
>	Indicates menu selection in a graphical user interface.		File > New
			File > New > Templates

Table 4 describes the shell prompt conventions used in this book.

Table 4   Shell Prompts

Shell	Prompt
C shell	*machine-name*%
C shell superuser	*machine-name*#
Bourne shell and Korn shell	$
Bourne shell and Korn shell superuser	#

Input and output of Directory Server commands are usually expressed using the
LDAP Data Interchange Format (LDIF) [RFC 2849]. Lines are wrapped for
readability.

Related Books

The following books can be found in HTML and PDF at http://www.sun.com/documentation/.

Directory Server Books

Directory Server Release Notes

Directory Server Technical Overview

Directory Server Deployment Planning Guide

Directory Server Installation and Migration Guide

Directory Server Performance Tuning Guide

Directory Server Administration Guide

Directory Server Administration Reference

Directory Server Plug-in Developer's Guide

Directory Server Plug-in Developer's Reference

Directory Server Man Page Reference

Administration Server Books

Administration Server Release Notes

Administration Server Administration Guide

Administration Server Man Page Reference

Directory Proxy Server Books

Directory Proxy Server Release Notes

Directory Proxy Server Administration Guide

Related Java Enterprise System Books

Java Enterprise System Installation Guide

Java Enterprise System Upgrade and Migration Guide

Java Enterprise System Glossary

Documentation, Support, and Training

Table 5 provides links to Sun documentation, support, and training information.

Table 5   Documentation, Support, and Training links

Typeface	Meaning	Examples
Documentation	`http://www.sun.com/documentation/`	Download PDF and HTML documents, and order printed documents.
Support and Training	`http://www.sun.com/supportraining/`	Obtain technical support, download patches, and learn about Sun courses.

Related Third-Party Web Site References

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. Use the web-based form to provide feedback to Sun:

http://www.sun.com/hwdocs/feedback/

Please provide the full document title and part number in the appropriate fields. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document. For example, the part number of this document is 817-7608-10.

Sun Welcomes Your Comments

Finding Installation and Upgrade Instructions

Directory Server 5.2 installs as a component of the Sun Java™ Enterprise System. For information about how to install Directory Server 5.2, refer to the *Java Enterprise System Installation Guide.*

For information about how to perform a patch upgrade of Directory Server from a previous version of the Java Enterprise System, refer to the *Java Enterprise System Upgrade and Migration Guide.*

This guide contains information about how to upgrade to Directory Server 5.2 from Directory Server 4.x, Directory Server 5.0, and Directory Server 5.1. This guide also contains information about how to use Directory Server in a Sun Cluster environment.

Overview of Upgrading

This chapter helps you determine whether to patch your Directory Server installation or perform full upgrade and migration from an earlier version of Directory Server. Earlier versions include Netscape Directory Server 4.x, iPlanet Directory Server 5.x, Sun ONE Directory Server 5.2, and Sun Java System Directory Server 200xQx. This chapter also helps you understand the full upgrade process, and identify attribute values you may need to set manually after performing full upgrade.

NOTE	This guide does not explain how to upgrade from Innosoft Distributed Directory Server 4.5.1.

This chapter includes the following sections:

- Patch Update or Upgrade?
- Before You Upgrade
- The Upgrade and Data Migration Process
- Migration for Specific Attribute Types

Patch Update or Upgrade?

Depending on the existing Directory Server version in use, you choose to:

- Patch update Directory Server, replacing software binaries and migrating certificate formats but not migrating directory data
- Upgrade Directory Server, installing new software and migrating directory data and server configuration to the new instance

NOTE	Directory Server remote administration depends on Administration Server. When you therefore patch update or upgrade Directory Server, you must also patch update or upgrade Administration Server for the same *ServerRoot.*

Decision to Update or to Upgrade

Table 2-1 helps you determine whether you may patch update, or must upgrade and migrate data.

Table 2-1   Patch Updating or Upgrading and Migrating Data to the Current Version

Existing Version	On Solaris Systems	On Other Platforms
4.x, 5.0, 5.1	*Upgrade and migrate*	*Upgrade and migrate*
5.2 packages (SUNWds*) including 5.2 2003Q4	*Patch update*	N/A
5.2 packages (SUNWds*) including 5.2 2004Q2	*Patch update*	N/A
5.2 compressed archive (*all platforms*)[1]	*Patch update*	*Patch update*

1. Refer to the *Directory Server 5.2 Release Notes* for details. The updates for 5.2 compressed archive versions contain essentially the same fixes and enhancements as those provided in the packaged version.

For instructions on performing a patch update, see the *Java Enterprise System Upgrade and Migration Guide.*

For instructions on performing upgrade and migration, read the rest of this chapter then refer to Chapter 3, "Upgrading and Migrating Data for a Single Directory Server Installation" on page 37.

Data Service Interruption When Patch Updating on Sun Cluster

When patch updating Directory Server running as a data service on Sun Cluster, you must stop both the Directory Server and associated Administration Server data services during the patch update process. If you leave the data services running during and the service fails over to a node not running the same version of the software, you run the risk of corrupting directory data.

Before You Upgrade

This section provides an overview of upgrade and data migration. Before upgrading, familiarize yourself with the new features and fixes available in the current version. Take the opportunity to review design decisions made during implementation of existing directory services.

When Upgrading a Single Server Instance

Upgrading a server instance involves installing the new server alongside the existing server in a different installation location, using a different *serverID* and a different port number, stopping the old server, migrating configuration and directory data, and then having clients make requests to the new server.

NOTE	Ensure you have sufficient disk space on the host where you run the existing server. The upgrade process requires at least enough *local disk space* to house binaries and databases for both the old and new servers, and also enough extra space to hold LDIF files containing the entries in all existing suffixes. You may estimate the local disk space required as somewhat larger than:

```
2 * (space for existing server) + (space for LDIF files)
```

The upgrade process must be performed with both servers on the same host, as data migration cannot be performed over networked drives.

Handling a Central Configuration Directory Server

If you have a separate, central Directory Server maintaining configuration data for your server topology (o=NetscapeRoot suffix), then that server needs to remain available during upgrade of all other servers. You might for example upgrade the configuration directory server before all other servers in your topology.

When Upgrading Multiple Replicated Servers

Not surprisingly, upgrading multiple servers involves upgrading each server individually. The *order* in which you upgrade servers depends, however, on the software version of existing servers and on the replication topology.

For upgrades from 5.x, perform the upgrade in the following order:

1. Upgrade the consumers.

2. Upgrade the hubs

3. Upgrade the masters.

Refer to "Example 5.x Upgrade Scenario" on page 52 for how you might do this in a particular instance.

For upgrades from 4.x, perform the upgrade in the following order:

1. Upgrade the 4.x master

2. Upgrade each branch of consumers being replicated from the master, starting with the consumer closest to the master in terms of replication.

Refer to "Example 4.x Upgrade Scenario" on page 44 for how you might do this in a particular instance.

If the existing environment involves multiple, replicated servers, read all relevant sections of this chapter carefully before proceeding with the upgrade. You must plan your approach fully to avoid unnecessary downtime.

The Upgrade and Data Migration Process

Directory Server provides a script (`migrateInstance5`) to help you migrate data from previous versions to the current version. The migration script performs the following tasks in sequence:

1. Stops your old server, and backs up its current configuration.

2. Checks schema configuration files, notifying you of differences between the standard schema configuration files and those used by your old server.

   (From 4.x only) If a 4.x server uses custom schema not installed in the default location, under *ServerRoot*/slapd-*serverID*/config, you must adjust the configuration manually before migrating directory data.

3. Creates a database for each suffix stored in the old server.

   (From 4.x only) 4.x servers supported multiple suffixes per database. The migration script creates a database for each suffix on the new server.

4. Migrates server and database configuration parameters.

   4.x servers store such parameters in `slapd.conf`. 5.x servers store such parameters as entries in `dse.ldif`.

NOTE   The script does not migrate data under `o=NetscapeRoot`.

When deploying a server such as Application Server, Identity Server, or Messaging Server that rely on data in this suffix, migrate the data in `o=NetscapeRoot` manually, or using tools provided with the server in question.

(From 4.x only) The migration script does not migrate all 4.x server parameters. In some cases, you must migrate 4.x attribute values manually. Refer to "Migration for Specific Attribute Types" on page 22 for details.

5. Migrates user-defined schema objects.

6. Migrates indexes.

7. Migrates the configuration of standard server plug-ins, including:

   ❍ 7-bit Check

   ❍ Binary Syntax

   ❍ Case Exact String Syntax

   ❍ Case Ignore String Syntax

   ❍ Distinguished Name Syntax

   ❍ Integer Syntax

   ❍ Internationalization

   ❍ Retro Changelog

   ❍ Referential Integrity (see note that follows)

   ❍ Telephone Syntax

❍ UID Uniqueness

If you have *changed* the configuration of a plug-in not in the list, you must fix that configuration manually after running the script.

You must also migrate all custom plug-ins manually. At minimum, you must recompile all custom plug-ins and add their configuration to the directory. Refer to the *Directory Server Plug-in Developer's Guide* for a detailed list of plug-in API changes.

NOTE	When replicating from a 4.x master to a 5.x consumer with referential integrity enabled, you must reconfigure the referential integrity plug-in on the 4.x master to write referential integrity changes to the 4.x changelog. Refer to the *Directory Server Administration Guide* for further details.

8. (From 5.x only) Migrates replication agreements.

NOTE	Before replicating from a 5.2 Directory Server to a 5.1 server, set `nsslapd-schema-repl-useronly` on `cn=config` to `on`. Otherwise the 5.2 schema are pushed to the 5.1 server, preventing the 5.1 server from restarting due to duplicate objects.

You must, however, reinitialize replicas after migrating the data.

9. Migrates the certificate database, and SSL parameters.

10. (From 5.x only) Migrates database links.

11. (From 5.x only) Migrates replication entries.

12. Migrates the SNMP configuration.

After the migration script completes, clients may send requests to the new server.

Migration for Specific Attribute Types

This section covers attribute types whose values are migrated by the `migrateInstance5` script. It identifies which attributes are migrated automatically by the migration script, and which ones must be set manually.

For migration from 4.x, it also describes the mapping of configuration parameters to configuration attributes and configuration entries in the current format.

Migrating From Directory Server 4.x

In Directory Server 4.x architecture, all configuration parameters were stored in text files. In Directory Server 5.x, all configuration attributes are stored in LDAP configuration entries in the dse.ldif file.

This section describes the mapping of configuration parameters in Directory Server 4.x to the corresponding LDAP configuration entries and attributes in Directory Server 5.2.

Server Attributes

In Directory Server 4.x, configuration parameters are stored in the slapd.conf file under the /usr/netscape/server4/slapd-*serverID* directory.

The corresponding configuration attributes in the current version are stored in the cn=config entry. Table 2-2 shows the mapping of Directory Server 4.x configuration parameters to current configuration attributes.

Table 2-2   Mapping of Legacy Server Parameters to Configuration Attributes

Legacy Configuration Parameter	Current Configuration Attribute
(No equivalent)	nsslapd-depends-on-named
(No equivalent)	nsslapd-depends-on-type
accesscontrol	nsslapd-accesscontrol
accesslog	nsslapd-accesslog
accesslog-level	nsslapd-accesslog-level
accesslog-logexpirationtime	nsslapd-accesslog-logexpirationtime
accesslog-logexpirationtimeunit	nsslapd-accesslog-logexpirationtimeunit
accesslog-logrotationtime	nsslapd-accesslog-logrotationtime
accesslog-logrotationtimeunit	nsslapd-accesslog-logrotationtimeunit
accesslog-maxlogdiskspace	nsslapd-accesslog-logmaxdiskspace
accesslog-maxlogsize	nsslapd-accesslog-maxlogsize
accesslog-MaxNumOfLogsPerDir	nsslapd-accesslog-maxlogsperdir

Table 2-2   Mapping of Legacy Server Parameters to Configuration Attributes *(Continued)*

Legacy Configuration Parameter	Current Configuration Attribute
accesslog-minfreediskspace	nsslapd-accesslog-minfreediskspace
auditfile	nsslapd-auditlog
audit-logging-enabled	nsslapd-audit-logging-enabled
auditlog-logexpirationtime	nsslapd-auditlog-logexpirationtime
auditlog-logexpirationtimeunit	nsslapd-auditlog-logexpirationtimeunit
auditlog-logrotationtime	nsslapd-auditlog-logrotationtime
auditlog-logrotationtimeunit	nsslapd-auditlog-logrotationtimeunit
auditlog-maxlogdiskspace	nsslapd-auditlog-logmaxdiskspace
auditlog-maxlogsize	nsslapd-auditlog-maxlogsize
auditlog-MaxNumOfLogsPerDir	nsslapd-auditlog-maxlogsperdir
auditlog-minfreediskspace	nsslapd-auditlog-minfreediskspace
certmap-basedn	nsslapd-certmap-basedn
enquote_sup_oc	nsslapd-enquote_sup_oc
errorlog	nsslapd-errorlog
error-logging-enabled	nsslapd-error-logging-enabled
errorlog-logexpirationtime	nsslapd-errorlog-logexpirationtime
errorlog-logexpirationtimeunit	nsslapd-errorlog-logexpirationtimeunit
errorlog-logrotationtime	nsslapd-errorlog-logrotationtime
errorlog-logrotationtimeunit	nsslapd-errorlog-logrotationtimeunit
errorlog-maxlogdiskspace	nsslapd-errorlog-logmaxdiskspace
errorlog-maxlogsize	nsslapd-errorlog-maxlogsize
errorlog-maxlogsperdir	nsslapd-errorlog-maxlogsperdir
errorlog-minfreediskspace	nsslapd-errorlog-logminfreediskspace
idletimeout	nsslapd-idletimeout
instancedir	nsslapd-instancedir
ioblocktimeout	nsslapd-ioblocktimeout
lastmod	nsslapd-lastmod
listenhost	nsslapd-listenhost

Table 2-2   Mapping of Legacy Server Parameters to Configuration Attributes *(Continued)*

Legacy Configuration Parameter	Current Configuration Attribute
localhost	nsslapd-localhost
localuser	nsslapd-localuser
logbuffering	nsslapd-accesslog-buffering
loglevel	nsslapd-infolog-area
maxbersize	nsslapd-maxbersize
maxdescriptors	nsslapd-maxdescriptors
maxthreadsperconn	nsslapd-maxthreadsperconn
nagle	nsslapd-nagle
port	nsslapd-port
pw_change	passwordChange
pw_exp	passwordExp
pw_inhistory	passwordinHistory
pw_lockduration	passwordLockoutDuration
pw_lockout	passwordLockout
pw_maxage	passwordMaxAge
pw_maxfailure	passwordMaxFailure
pw_minage	passwordMinAge
pw_minlength	passwordMinLength
pw_must_change	passwordMustChange
pw_reset_failurecount	passwordResetFailureCount
pw_storagescheme	passwordStorageScheme
pw_syntax	passwordCheckSyntax
pw_unlock	passwordUnlock
pw_warning	passwordWarning
referral	nsslapd-referral
reservedescriptors	nsslapd-reservedescriptors
result_tweak	nsslapd-result_tweak
return_exact_case	nsslapd-return_exact_case

Table 2-2   Mapping of Legacy Server Parameters to Configuration Attributes *(Continued)*

Legacy Configuration Parameter	Current Configuration Attribute
rootdn	nsslapd-rootdn
rootpw	nsslapd-rootpw
rootpwstoragescheme	nsslapd-rootpwstoragescheme
schemacheck	nsslapd-schemacheck
secure-port	nsslapd-securePort
security	nsslapd-security
sizelimit	nsslapd-sizelimit
SSL3ciphers	nsslapd-SSL3ciphers
threadnumber	nsslapd-threadnumber
timelimit	nsslapd-timelimit

Database Attributes

In Directory Server 4.x, database parameters are stored in the `slapd.ldbm.conf` file under the `/usr/netscape/server4/slapd-`*serverID* directory.

Because one instance of Directory Server 5.x can manage several databases, the corresponding attributes in Directory Server 5.x are stored in a general entry for all databases (`cn=config,cn=ldbm database,cn=plugins,cn=config`), or in an entry specific to a particular database, of the form

`cn=`*database instance name*`,cn=ldbm database,cn=config`

Table 2-3 shows the mapping of general database configuration parameters between Directory Server 4.x and Directory Server 5.2.

Table 2-3   Mapping of General Legacy Database Parameters to Configuration Attributes

Legacy Configuration Parameter	Current Configuration Attribute
allidsthreshold	nsslapd-allidsthreshold
database	OBSOLETE (used to specify database type)
lookthroughlimit	nsslapd-lookthroughlimit
mode	nsslapd-mode

Table 2-4 shows the mapping of database-specific parameters between Directory Server 4.x and Directory Server 5.2.

Table 2-4   Mapping of Database-Specific Legacy Parameters to Configuration Attributes

Legacy Configuration Parameter	Current Configuration Attribute
cachesize	nsslapd-cachesize
directory	nsslapd-directory
readonly	nsslapd-readonly

Not all parameters are migrated by the migrateInstance5 script. Table 2-5 indicates Directory Server 4.x parameters that are not migrated automatically, and why automatic migration is not done in each case.

Table 2-5   Legacy Parameters Not Migrated by the Migration Script

Legacy Configuration Parameter	Current Configuration Attribute	Reason
accesslog	nsslapd-accesslog	Set up automatically. Path name of the database access log.
auditfile	nsslapd-auditlog	Set up automatically. Path name of the log used to record changes made to the database.
database	(No equivalent)	OBSOLETE (used to specify database type)
directory	nsslapd-directory	Set up during installation.
errorlog	nsslapd-errorlog	Set up automatically. Path name of the log used to record error messages generated by Directory Server.
instancedir	nsslapd-instancedir	Set up during installation.
localhost	nsslapd-localhost	Already configured.
port	nsslapd-port	Configured manually during installation.
pw_history	(No equivalent)	OBSOLETE
result_tweak	nsslapd-result_tweak	Reserved for future use. Do not use, change or remove. Doing so may have unpredictable results.
rootdn	nsslapd-rootdn	Configured manually during installation.
rootpw	nsslapd-rootpw	Configured manually during installation.

Table 2-6 indicates the parameters that are migrated but are potentially problematic. You are advised to check their values in the new installation:

Table 2-6   Legacy Parameters Migrated by the Migration Script

Legacy Configuration Parameter	Directory Server 5.2 Configuration Attribute
maxbersize	nsslapd-maxbersize
maxthreadsperconn	nsslapd-maxthreadsperconn
nagle	nsslapd-nagle
return_exact_case	nsslapd-return_exact_case
threadnumber	nsslapd-threadnumber

Migrating From Directory Server 5.x

All versions of Directory Server 5.x store configuration information in the same way. This section explains which configuration attributes are automatically migrated by the migrateInstance5 script, and which ones are not. Attributes which are not automatically migrated are either configured during the installation process for the new Directory Server, or need to be configured manually for security reasons after the initial setup.

Global Server Configuration Attributes

The following list provides the configuration attributes stored in the cn=config entry that are automatically migrated when you run the migrateInstance5 script:

* nsslapd-accesscontrol

* nsslapd-accesslog-level

* nsslapd-accesslog-logbuffering

* nsslapd-accesslog-logexpirationtime

* nsslapd-accesslog-logexpirationtimeunit

* nsslapd-accesslog-logging-enabled

* nsslapd-accesslog-logmaxdiskspace

* nsslapd-accesslog-logminfreediskspace

* nsslapd-accesslog-logrotationtime

* nsslapd-accesslog-logrotationtimeunit

- `nsslapd-accesslog-maxlogsize`

- `nsslapd-accesslog-maxlogsperdir`

- `nsslapd-attribute_name_exceptions`

- `nsslapd-auditlog-logexpirationtime`

- `nsslapd-auditlog-logexpirationtimeunit`

- `nsslapd-auditlog-logging-enabled`

- `nsslapd-auditlog-logmaxdiskspace`

- `nsslapd-auditlog-logminfreediskspace`

- `nsslapd-auditlog-logrotationtime`

- `nsslapd-auditlog-logrotationtimeunit`

- `nsslapd-auditlog-maxlogsize`

- `nsslapd-auditlog-maxlogsperdir`

- `nsslapd-certmap-basedn`

- `nsslapd-ds4-compatible-schema`

- `nsslapd-enquote_sup_oc`

- `nsslapd-errorlog-level`

- `nsslapd-errorlog-logexpirationtime`

- `nsslapd-errorlog-logexpirationtimeunit`

- `nsslapd-errorlog-logging-enabled`

- `nsslapd-errorlog-logmaxdiskspace`

- `nsslapd-errorlog-logminfreediskspace`

- `nsslapd-errorlog-logrotationtime`

- `nsslapd-errorlog-logrotationtimeunit`

- `nsslapd-errorlog-maxlogsize`

- `nsslapd-errorlog-maxlogsperdir`

- `nsslapd-groupevalnestlevel`

- `nsslapd-idletimeout`

- `nsslapd-ioblocktimeout`

- `nsslapd-lastmod`

- `nsslapd-listenhost`

- `nsslapd-maxdescriptors`

- `nsslapd-nagle`

- `nsslapd-plugin-depends-on-name`

- `nsslapd-plugin-depends-on-type`

- `nsslapd-readonly`

- `nsslapd-referral`

- `nsslapd-referralmode`

- `nsslapd-reservedescriptors`

- `nsslapd-rootpwstoragescheme`

- `nsslapd-schemacheck`

- `nsslapd-securePort`

- `nsslapd-security`

- `nsslapd-sizelimit`

- `nsslapd-SSL3ciphers`

- `nsslapd-timelimit`

NOTE   The attribute `nsslapd-errorlog-level` has been deprecated. It is still supported for backward compatibility but has been replaced by the `nsslapd-infolog-area` and `nsslapd-infolog-level` attributes.

Table 2-7 lists the configuration attributes stored in the `cn=config` entry that are *not* automatically migrated when you run the `migrateInstance5` script. Attributes that are not automatically migrated are either configured during the installation process for the new Directory Server, or need to be configured manually. The reason for not migrating an attribute is stated in the table.

Table 2-7   Attributes in `cn=config` Not Migrated

Attribute Name	Reason for not Migrating Automatically
`nsslapd-accesslog`	Path name to the log that records database access. It is set up during installation.
`nsslapd-accesslog-list`	Read-only attribute.
`nsslapd-auditlog`	Path name to the log that records changes made to the directory database. It is set up during installation.
`nsslapd-errorlog`	Path name to the log that records error messages generated by Directory Server. It is set up during installation.
`nsslapd-errorlog-list`	Read-only attribute.
`nsslapd-instancedir`	Configured during the installation process.
`nsslapd-localhost`	Already set up.
`nsslapd-localuser`	Configured during the installation process.
`nsslapd-maxbersize`	Do not change the value of this attribute unless told to do so by Sun technical staff.
`nsslapd-maxthreadsperconn`	This attribute corresponds to a system parameter.
`nsslapd-plugin`	Internal computed attribute.
`nsslapd-port`	Configured during the installation process.
`nsslapd-result-tweak`	Reserved for future use. Do not change or remove.
`nsslapd-return-exact-case`	Do not modify unless you have legacy client applications that can check the case of attribute names in results returned from the server.
`nsslapd-rootdn`	Configured during the installation process.
`nsslapd-rootpw`	Configured during the installation process.
`nsslapd-threadnumber`	This attribute is not available from Directory Server Console.

Password Policy Attributes

The attributes that determine the password policy are stored in the entry `cn=Password Policy,cn=config`. Note that the location of these attributes has changed. In previous versions of Directory Server, they were located directly under `cn=config`. The following list provides the password policy attributes that are automatically migrated when you run the `migrateInstance5` script:

- `passwordChange`

- `passwordCheckSyntax`

- `passwordExp`

- `passwordExpireWithoutWarning`

- `passwordInHistory`

- `passwordLockout`

- `passwordLockoutDuration`

- `passwordMaxAge`

- `passwordMaxFailure`

- `passwordMinAge`

- `passwordMinLength`

- `passwordMustChange`

- `passwordResetFailureCount`

- `passwordStorageScheme`

- `passwordUnlock`

- `passwordWarning`

Database Attributes

All general database configuration attributes are automatically migrated. These attributes are stored in the entry `cn=config,cn=ldbm database, cn=plugins,cn=config,` and are as follows:

- `nsslapd-allidthreshold`

- `nsslapd-cache-autosize`

- `nsslapd-cache-autosize-split`

- `nsslapd-dbcachesize`

- `nsslapd-db-transaction-logging`

- `nsslapd-lookthroughlimit`

- `nsslapd-mode`

Database-specific attributes are stored in entries of the form cn=*database instance name*,cn=ldbm database,cn=config. The following list provides the attributes that are migrated:

- nsslapd-cachememsize
- nsslapd-cachesize
- nsslapd-readonly
- nsslapd-require-index

Table 2-8 lists the attributes that are *not* migrated automatically and indicates why this is the case.

Table 2-8   Database-Specific Attributes Not Migrated

Attribute Name	Reason For Not Migrating Automatically
nsslapd-db-checkpoint-interval	This attribute is provided only for system modification/diagnostics and should be changed only under guidance from Sun technical staff. Inconsistent settings of this attribute might cause Directory Server crashes.
nsslapd-db-durable-transactions	This attribute is provided only for system modification/diagnostics and should be changed only under guidance from Sun technical staff. Inconsistent settings of this attribute might cause Directory Server crashes.
nsslapd-db-home-directory	If you have several Directory Servers running on the same machine, the value of this attribute must be different for each instance of Directory Server. Therefore, it needs to be configured manually.
nsslapd-db-logdirectory	Set up automatically during installation.
nsslapd-directory	Set up automatically during installation.

Chained Suffix Attributes

All chained suffix configuration attributes are migrated automatically. The following configuration attributes are common to all chained suffixes. These attributes are stored in the entry cn=config,cn=chaining database, cn=plugins,cn=config.

- nsActivechainingComponents
- nsTransmittedControls

The following configuration attributes apply to a default instance of a chained suffix. These attributes are stored in the entry `cn=default instance config, cn=chaining database,cn=plugins,cn=config`.

- `nsAbandonedSearchCheckInterval`

- `nsBindConnectionsLimit`

- `nsBindRetryLimit`

- `nsBindTimeout`

- `nsCheckLocalACI`

- `nsConcurrentBindLimit`

- `nsConcurrentOperationsLimit`

- `nsConnectionLife`

- `nsHopLimit`

- `nsmaxresponsedelay`

- `nsmaxtestresponsedelay`

- `nsOperationConnectionslimit`

- `nsProxiedAuthorization`

- `nsReferralOnScopedSearch`

- `nsslapd-sizelimit`

- `nsslapd-timelimit`

SNMP Attributes

All SNMP configuration attributes are automatically migrated. These attributes are stored in the entry `cn=SNMP,cn=config`, and are as follows:

- `nssnmpcontact`

- `nssnmpdescription`

- `nssnmpenabled`

- `nssnmplocation`

- `nssnmpmasterhost`

- `nssnmpmasterport`

- `nssnmporganization`

Upgrading and Migrating Data for a Single Directory Server Installation

This chapter describes the full upgrade process for converting a single existing server installation to the current version. Follow the instructions in the procedures described here only if you must perform full upgrade, as opposed to patch upgrade.

NOTE   If an existing 4.x server uses custom schema, ensure the migration script can find the custom schema before migrating any data. Read "(From 4.x) Handling Custom Schema" on page 38 for details.

If the migration script does not recognize the custom schema, it does not migrate the schema, but instead applies standard schema files after migrating the data to the new server. Applying the standard schema to entries that conform to custom schema may render modifications impossible, thereby making the upgraded directory read-only.

This chapter includes the following sections:

- Installing the New Server

- (From 4.x) Handling Custom Schema

- Migrating Existing Data

- (From 4.x) Creating Replication Agreements

- (Optional) Reusing the Existing Port Number

Installing the New Server

Follow the links in Chapter 1, "Finding Installation and Upgrade Instructions" on page 15 to determine how to install the new server on the same node as the existing server.

NOTE	Ensure you have a current backup of the existing server before installing the new server.

The new server must reside in a different installation location than the existing server. It must also be identified by a different *serverID*.

Although you may choose to reuse most of the configuration information supplied for the original installation, do not reuse the existing port number. Instead, you may change the port number of the new server after migrating existing data.

(From 4.x) Handling Custom Schema

The script provided for migrating data recognizes only those custom schema either placed in the standard `slapd.user_oc.conf` and `slapd.user_at.conf` files, or placed in other files and included in `slapd.conf` using `useroc` and `userat` directives. If, for example, custom schema are included directly in `slapd.at.conf` or `slapd.oc.conf`, the migration script does not recognize them.

Perform the following steps before proceeding with the upgrade.

1. Compare `slapd.at.conf` or `slapd.oc.conf` with the standard files provided under *ServerRoot*`/bin/slapd/install/version4/` of the new server, transcribing custom schema elements to `slapd.user_oc.conf` and `slapd.user_at.conf` files.

   If the custom object classes have inheritance relationships, ensure that superior object classes precede others in the schema configuration file.

2. If custom attributes were added to standard object classes in `slapd.oc.conf`, create a new object class including the attributes in `slapd.user_oc.conf`, and add the new object class to every entry in the existing directory that uses the custom attributes.

3. Include the `slapd.user_oc.conf` and `slapd.user_at.conf` files in the `slapd.conf` file for the existing server using the `useroc` and `userat` directives, placing the new directives adjacent to include statements for other files.

At this point, all custom schema used by the existing server should reside in
`slapd.user_oc.conf` or `slapd.user_at.conf`, and `slapd.conf` should include
the files using the `useroc` and `userat` directives.

Migrating Existing Data

After handling custom schema, perform the following steps to migrate existing
data to a new server.

1. If you intend to initialize replication on the new Directory Server offline from
   files, obtain the files before proceeding.

   Refer to the *Directory Server Administration Guide* for instructions on exporting
   Directory Server data.

2. Ensure the new Directory Server is running.

3. Work as a user having the right to start, stop, and run database export and
   import on both the old and new servers.

   For example, become super user.

4. Set environment variables as shown in Table 3-1.

Table 3-1   Environment Variables for Migration

Variable	Value
PATH	*ServerRoot*/bin/slapd/admin/bin:$PATH
PERL5LIB	*ServerRoot*/bin/slapd/admin/bin

5. Run the migration script under the new server instance:

```
# cd ServerRoot/bin/slapd/admin/bin
# perl migrateInstance5 -p port52 -D "cn=directory manager" -w password -o oldServ -n newServ
```

Here, *oldServ* represents the full path to the old server instance, such as
`/usr/iplanet/servers/slapd-ldap` or `/usr/iplanet/ds5/slapd-ldap`, and
newServ represents the full path to the new server instance, such as
`/var/opt/mps/serverroot/slapd-dirserv`.

The script generates output as it proceeds. You may choose to redirect this output
to a file for review after migration completes.

Retire the old server only after migrating existing data to the new server.

(From 4.x) Creating Replication Agreements

If an existing 4.x server is involved in replication, upgrading involves recreating replication agreements after migrating data. Read Chapter 4, "Migrating a Replicated 4.x Topology" on page 41 before proceeding with the upgrade process.

Refer to the *Directory Server Administration Guide* for instructions on configuring replication for 5.2 servers.

(Optional) Reusing the Existing Port Number

After migrating data from the old server to the new, you may choose to retire the old server and have the new server listen on the same port as the old. Using the same port may allow client applications to continue operating without changing their configurations.

Refer to the *Directory Server Administration Guide* for instructions on changing the server port. Be sure to stop the old server before the new server starts to listen on the old port.

Migrating a Replicated 4.x Topology

When upgrading replicated 4.x servers, start by replicating to a new master, and then proceed branch by branch through the replication topology. This approach limits the volume of server synchronization traffic.

NOTE	Refer to the *Directory Server Administration Guide* for detailed instructions concerning replication configuration and initialization.

This chapter includes the following sections:

- Preparing the New Master

- Upgrading the Consumers

- Upgrading a Branch

- Adding Additional Servers

- Example 4.x Upgrade Scenario

Preparing the New Master

During the upgrade, the 5.2 server is configured as a master but functions as a legacy consumer in the 4.x topology. After the upgrade, the 4.x consumer capability is disabled, and the new server functions as a master in the 5.2 topology.

This procedure calls for manual configuration of the new master server. You may therefore install the new master on a different host than the existing master.

1. Follow the links in Chapter 1, "Finding Installation and Upgrade Instructions" on page 15 to determine how to install the new server.

2. Manually reproduce the configuration of the 4.x master on the new server.

3. Make the new server a master (for the 5.2 topology).

   Refer to the *Directory Server Administration Guide* for instructions.

4. Make the new server a legacy consumer of the 4.x master (for the 4.x topology).

   Again, refer to the *Directory Server Administration Guide* for instructions.

5. Initialize replication from the 4.x master to the new server.

   The process is described in "*Managing Replication*" in the *Netscape Directory Server Administration Guide*. Refer to the section entitled, "*Manual Consumer Initialization.*"

You may now upgrade the consumers.

Upgrading the Consumers

This procedure outlines the approach. Refer to subsequent procedures for details.

1. Upgrade all branches in the 4.x topology.

2. Add additional servers to the 5.2 topology as required.

3. Disable the legacy consumer agreement on the new master to sever the new topology from the old.

Upon completion of this procedure, the update process is complete.

Upgrading a Branch

Think of the existing 4.x replication topology as a tree with the master as the root element. Here, a *branch* denotes a set of replicated servers in that tree for which the flow of replication originates at the root node supplier, continues out through consumers in the midst of the tree, and finally arrives at leaf node consumer servers.

Upgrading a branch consists of replacing all old servers in the branch with new servers, working from the top down.

NOTE	While you upgrade a server, replication flow stops to all downstream servers in the branch. Consider redirecting client requests to another branch during the upgrade.

1. Proceed according to the instructions under Chapter 3, "Upgrading and Migrating Data for a Single Directory Server Installation" on page 37 or follow the links in Chapter 3, "Finding Patch Update Instructions" on page 37 to upgrade the top server in the branch.

   This cuts replication flow to the branch, temporarily bringing replication updates on downstream servers in the branch to a halt.

2. Configure the replication agreement on the new server in the 5.2 branch to receive updates from a 5.2 server closer in the replication topology to the new master.

   For example, configure the top server in the new branch to receive updates from the 5.2 master.

3. Initialize replication from the 5.2 supplier to the new 5.2 server.

   Depending on network capacity and volume of directory data compared to updates, offline initialization may be faster than online initialization.

4. Apply Step 1, Step 2, then Step 3 recursively along the branch until you have completed the steps for all leaf consumers.

Refer to the *Directory Server Administration Guide* for instructions on configuring replication agreements and initializing replication.

At this point, the update process is complete for the branch. Repeat the procedure for the remaining 4.x branches.

Adding Additional Servers

After completing the upgrade from the 4.x topology to the 5.2 topology, you may add additional masters, hubs, and consumers as required for the new topology.

Perform the following steps for each additional server.

1. Follow the links in Chapter 1, "Finding Installation and Upgrade Instructions" on page 15 to determine how to install the new server.

2. Adjust replication agreements on the new server to fit the planned topology.

   Refer to the *Directory Server Administration Guide* for instructions.

3. Initialize replication on the new server.

   Again, refer to the *Directory Server Administration Guide* for instructions.

Example 4.x Upgrade Scenario

Example 4.x Upgrade Scenario

Consider an upgrade for a 4.x master replicating to two branches, one with single consumer, one with hub supplying two consumers. This section shows the steps performed to upgrade to a new multi-master topology.

NOTE	In the figures that follow, solid arrows represent active replication. Dotted and dashed arrows represent interrupted replication. Dashed arrows alone represent instances being migrated.

Figure 4-1 shows the 4.x topology before upgrade.

Figure 4-1   Existing 4.x Topology Example
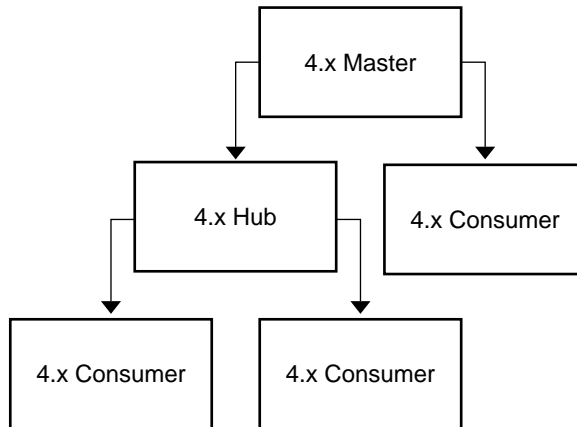
Figure 4-2 shows the addition of a 5.2 master that also functions as a legacy consumer of the 4.x master.

Example 4.x Upgrade Scenario

Figure 4-2   Example 4.x Topology with Additional New Server
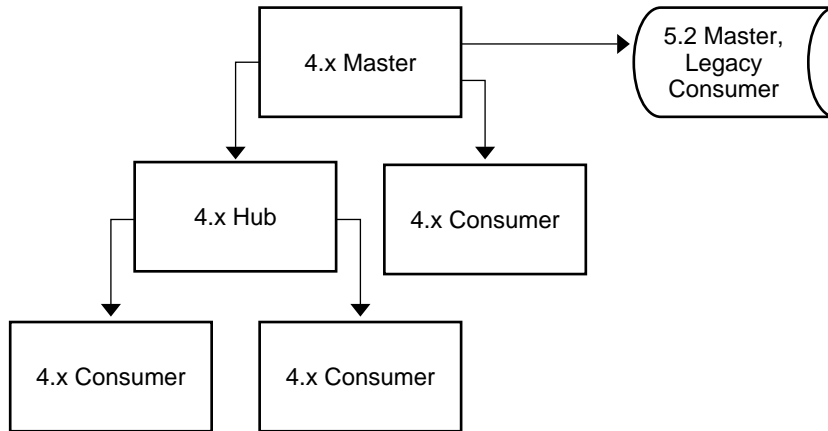
Figure 4-3 shows the first step in replacing a 4.x branch.

Notice the entire branch stops receiving replication updates during the upgrade. This interruption starts when the upstream 4.x consumer is stopped for upgrade, and ends when you restart the 4.x consumer.

As mentioned in the instructions, you may choose to direct client requests to consumers on another branch if clients require the very latest updates available.

Figure 4-3   Example 4.x Branch During Upgrade - Step 1

Example 4.x Upgrade Scenario
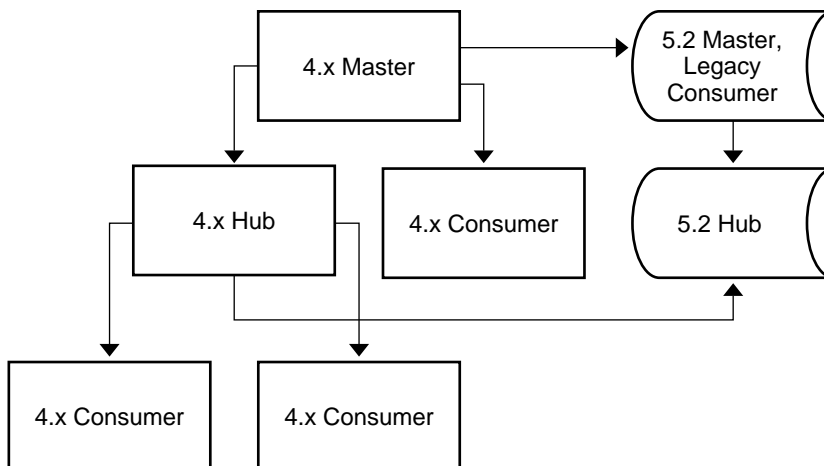
Figure 4-4 shows the next step in replacing a 4.x branch.

Figure 4-4   Example 4.x Branch During Upgrade - Step 2
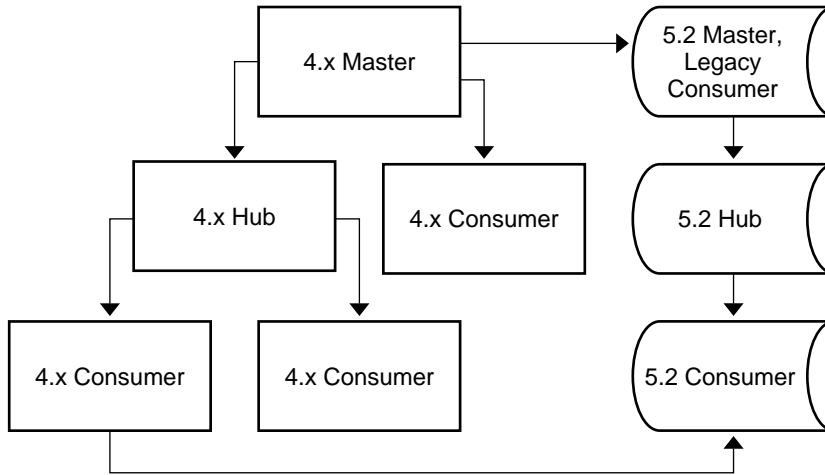
Figure 4-5 shows the next step in replacing a 4.x branch.

Figure 4-5   Example 4.x Branch During Upgrade - Step 3
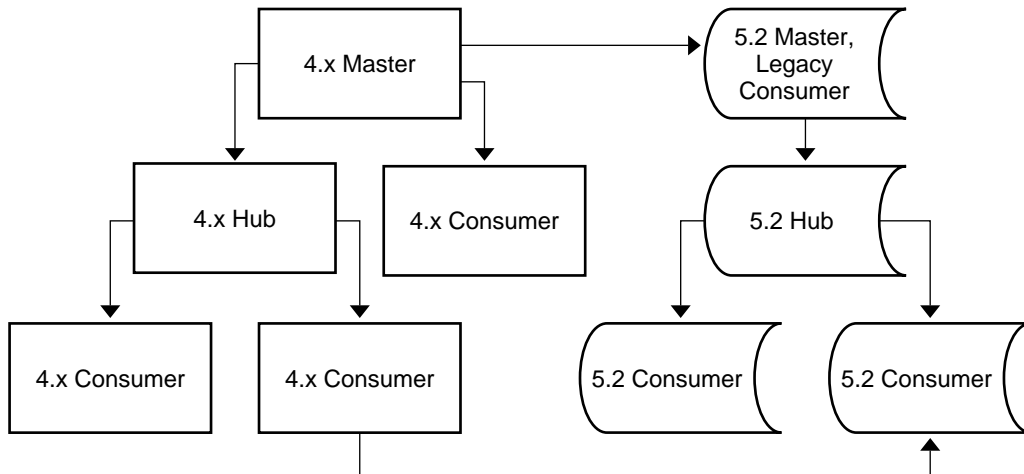
Figure 4-6 shows replacement of the other 4.x branch.

Example 4.x Upgrade Scenario

Figure 4-6   Example 4.x Branch During Upgrade - Next Branch
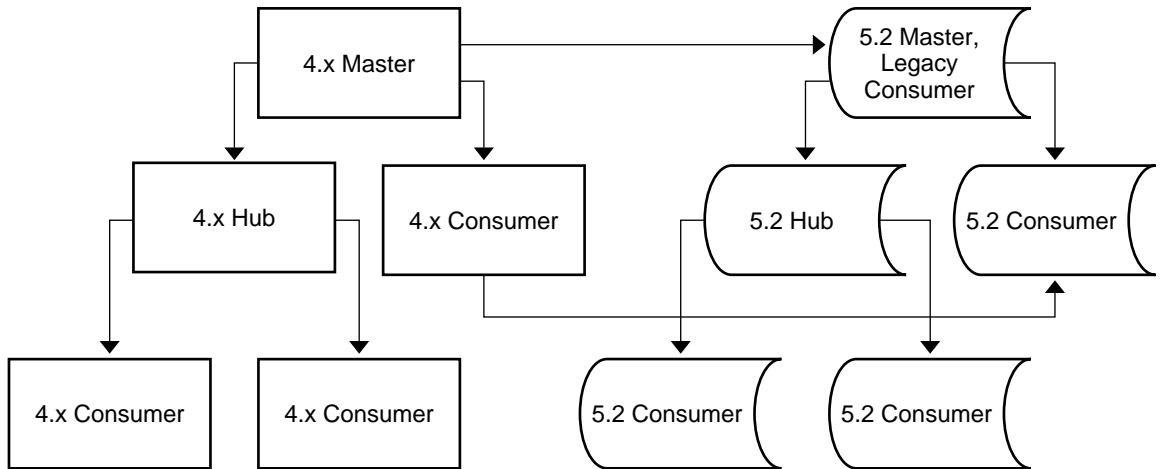
Figure 4-7 shows the two topologies side by side.

Figure 4-7   Example of 4.x and 5.2 Topologies During Upgrade
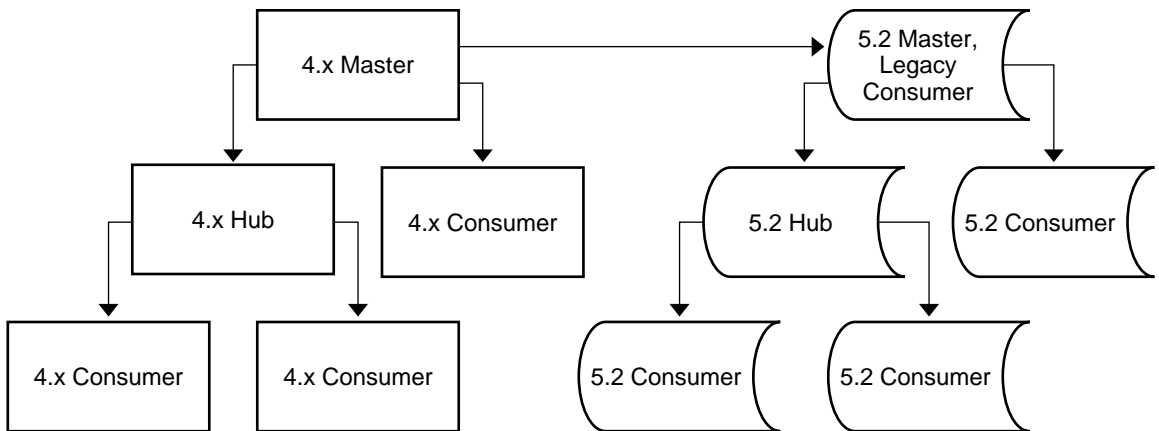
Figure 4-8 shows the addition of a master, a hub and additional replication agreements to the new topology.

Example 4.x Upgrade Scenario

Figure 4-8   Adding Servers to the 5.2 Topology
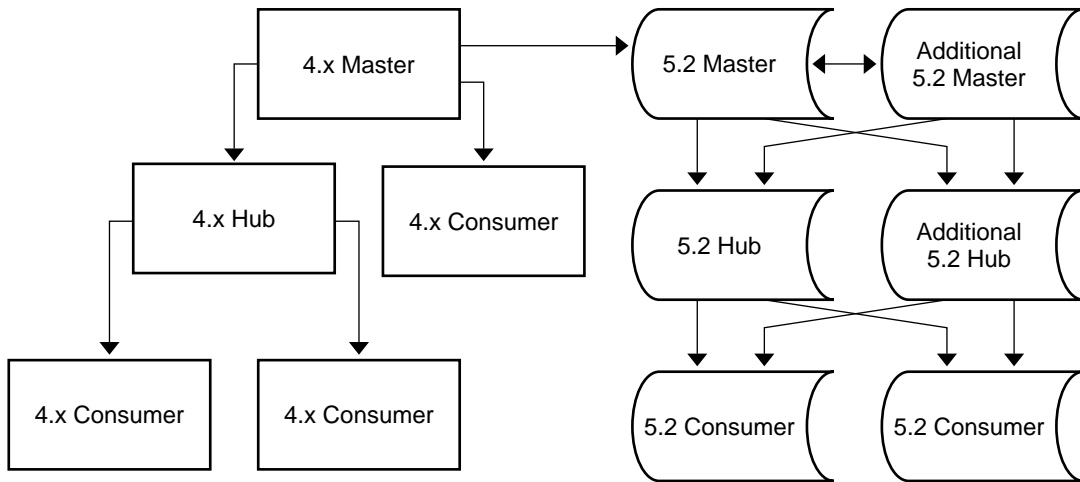
You may also add additional servers after completing the upgrade process.

Figure 4-9 shows removal of the replication agreement from the old 4.x master to the new 5.2 master.

Figure 4-9   Removing the Replication Agreement

Example 4.x Upgrade Scenario
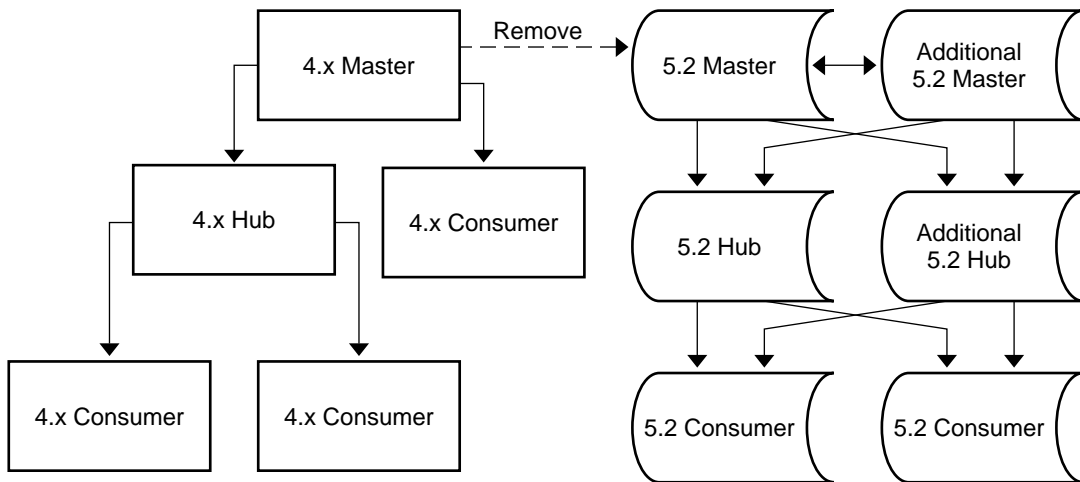
After redirecting client requests and removing the replication agreement, you may disable the 4.x servers. Figure 4-10 shows the resulting 5.2 topology.

Figure 4-10   Resulting 5.2 Topology
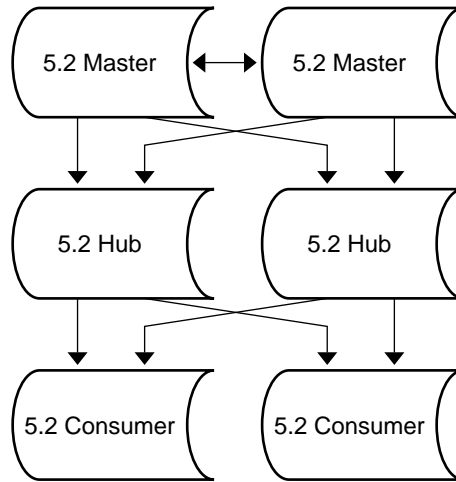
Client requests are now directed to the 5.2 topology.

Example 4.x Upgrade Scenario

Migrating a Replicated 5.x Topology

When upgrading replicated 5.x servers, you typically start with the consumers, continue with the hubs, and finish with the masters. This bottom-up approach involves interrupting only one server at a time, rather than interrupting an entire branch of the replication topology. The approach also helps you avoid potential custom schema synchronization issues between masters and consumers.

NOTE   The procedure described here applies the standard approach to upgrading a 5.x topology.

If, however, this bottom up approach fails to meet your specific requirements, then plan a different approach.

This chapter includes the following sections:

- Upgrading 5.x Servers
- Adding Additional Servers
- Example 5.x Upgrade Scenario

Upgrading 5.x Servers

1. For each consumer in the existing topology, proceed according to the instructions under Chapter 3, "Upgrading and Migrating Data for a Single Directory Server Installation" on page 37 or follow the links in Chapter 3, "Finding Patch Update Instructions" on page 37 to upgrade the consumer.

2. For each hub in the existing topology, proceed according to the same instructions to update the hub.

3. For each master in the existing topology, proceed according to the same instructions to update the master.

Adding Additional Servers

After completing the upgrade from the 5.x topology to the 5.2 topology, you may add additional masters, hubs, and consumers as required for the new topology.

Perform the following steps for each additional server.

1. Follow the links in Chapter 1, "Finding Installation and Upgrade Instructions" on page 15 to determine how to install the new server.

2. Adjust replication agreements on the new server to fit the planned topology.

3. Initialize replication on the new server.

Refer to the *Directory Server Administration Guide* for instructions on configuring replication agreements and initializing replication.

Upon completion of this procedure, the update process is complete. Clients may begin using servers in the upgraded replication topology.

Example 5.x Upgrade Scenario

Consider an upgrade for 5.x dual masters replicating to two hubs supplying two consumers. This section shows the steps performed to upgrade the topology to use 5.2 servers.

| NOTE | In the figures that follow, solid arrows represent active replication. Dotted and dashed arrows represent interrupted replication. Dashed arrows alone represent instances being migrated. |

Figure 5-1 shows the 5.x topology before the upgrade.

Example 5.x Upgrade Scenario

Figure 5-1   Existing 5.x Topology Example
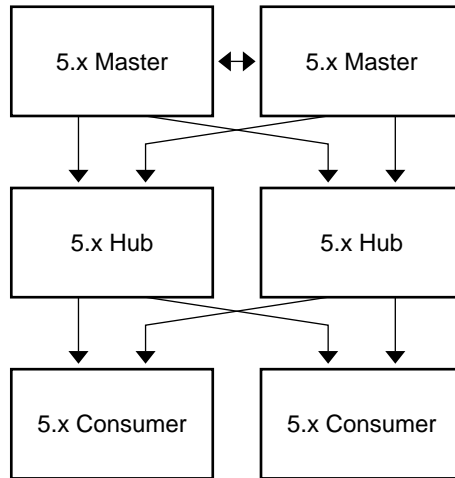
The first step involves upgrading consumers. Figure 5-2 shows the resulting topology.

Figure 5-2   Example 5.x Consumer Upgrade Step

Example 5.x Upgrade Scenario
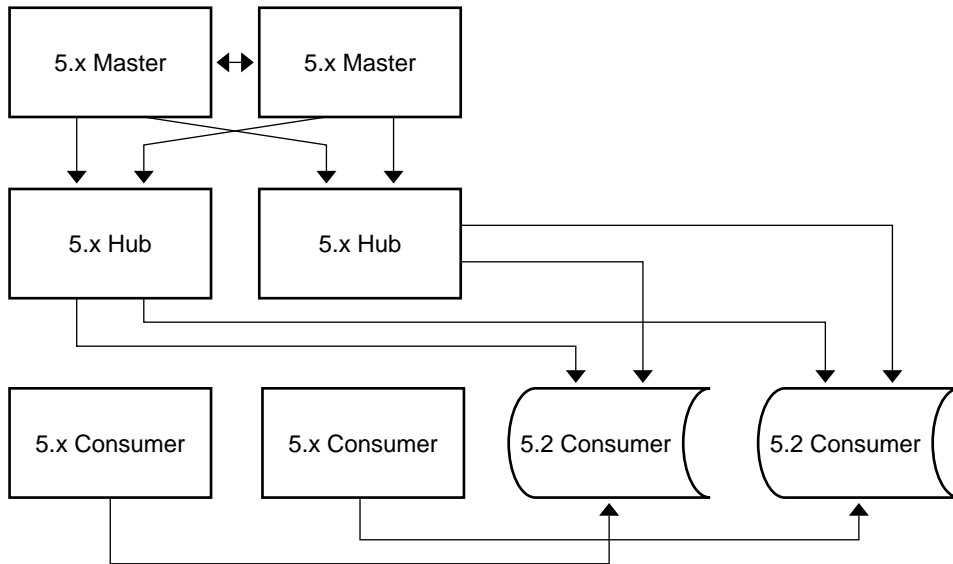
The next step involves upgrading hubs. Figure 5-3 shows the results.

Figure 5-3   Example 5.x Hub Upgrade Step
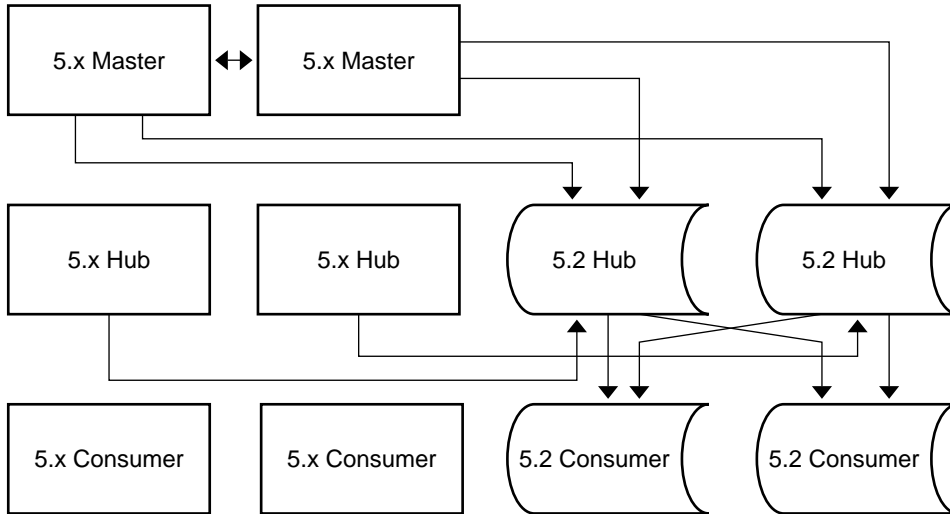
The next step involves upgrading masters. Figure 5-4 shows the results.

Figure 5-4   Example 5.x Master Upgrade - Step 3

Example 5.x Upgrade Scenario
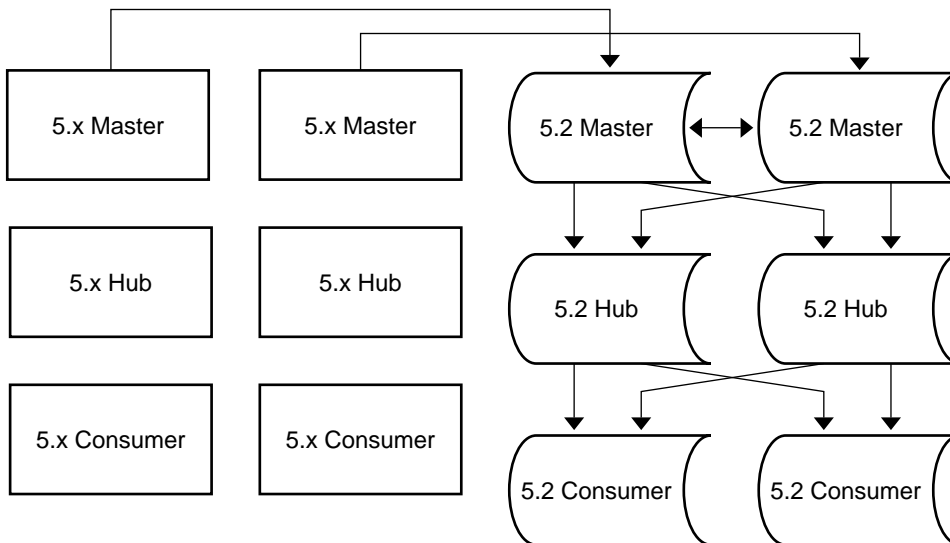
Figure 5-5 shows the 5.2 topology following the upgrade. At this point, servers in the old topology may be retired, and new servers added to the 5.2 topology.

Figure 5-5   Example 5.2 Topology after Upgrading
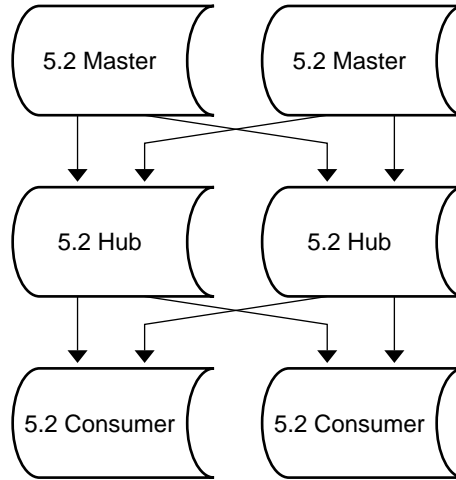
Client requests are now directed to the 5.2 topology.

Example 5.x Upgrade Scenario

Installing Sun Cluster HA for Directory Server

Currently, the only supported clustering technology for Directory Server is Sun Cluster 3.1, using the packaged versions of the products. Clustering is not supported for installations using compressed archive deliveries.

This appendix describes how to install and configure both the Sun Cluster HA for Directory Server data service and the associated Administration Server data service. This appendix also covers use of and upgrade to HA Storage Plus, which is currently the recommended storage resource type for use with Sun Cluster HA for Directory Server. Refer to the Sun Cluster 3.1 product documentation for Sun Cluster installation instructions and key concepts.

You must configure the data services as a failover services.

The following sections comprise this appendix:

- Before You Start
- Setting Up Network and File System Resources
- Installing the Servers
- Installing the Data Service Packages
- Configuring the Data Service
- Example Registration and Configuration
- Configuring Extension Properties
- Creating an Additional Directory Server Instance
- Uninstalling

Before You Start

Use this section in conjunction with the worksheets in the *Sun Cluster 3.1 Release Notes* as a checklist before performing installation and configuration.

Prior to starting your installation, consider these questions.

- Do you plan to run multiple Directory Server instances on the same node?

  It is recommended that you run all instances on the same node. If you do so, set nsslapd-listenhost on cn=config to the appropriate network resource (a logical host name, such as ds.example.com) as the IP address for each instance. Directory Server default behavior is to listen on all network interfaces.

- Do you run multiple data services in your Sun Cluster configuration?

  You may set up multiple data services in any order, with one exception: If you use Sun Cluster HA for DNS, you must set it up before setting up Sun Cluster HA for Directory Server.

NOTE   You must use the directoryserver(1M) and mpsadmserver(1M) commands and their subcommands when managing Directory Server and Administration Server from the command line.

Do not use the standalone scripts and binaries directly.

Table A-1 summarizes the Sun Cluster HA for Directory Server installation and configuration process.

Table A-1   Installation and Configuration Process

Task	What you should know
"Setting Up Network and File System Resources" on page 59 and "Installing the Servers" on page 62	The names of the cluster nodes that can master the data services.
	The logical host name to be used by clients accessing Directory Server such as ds.example.com. Refer to the Sun Cluster 3.1 product documentation for instructions on setting up a logical host name.
	The *ServerRoot* location on the failover file system such as /shared/ds where you place Directory Server data. This file system must be on a shared partition on shared disks. Note that packages must be installed on local file systems.

Table A-1   Installation and Configuration Process *(Continued)*

Task	What you should know
"Installing the Data Service Packages" on page 64	The `SUNWdsha` and `SUNWasha` packages provide the management interface for the data services so you can manage Directory Server and Administration Server with the same tools as other data services in the cluster.
"Configuring the Data Service" on page 64	The resource type names for Directory Server data service, `SUNW.dsldap`, and for the Administration Server data service, `SUNW.mps`.
	The names of the cluster nodes that can master the data services.
	The logical host name used by clients accessing Directory Server and Administration Server.
	The *ServerRoot* location on the file system where you install Directory Server data.
	The port on which Directory Server listens for client requests.
	The port on which Administration Server listens for client requests.
	The resource names defined in "Setting Up Network and File System Resources" on page 59.

Setting Up Network and File System Resources

Sun Cluster software manages one or more logical host names that differ both from node names and from host names for individual network interfaces. A clustered Directory Server instance typically relies on a single logical host name. Figure A-1 shows how a logical host name, managed by a two-node cluster, is not permanently associated with either of the nodes.

Figure A-1   Cluster with Two Nodes
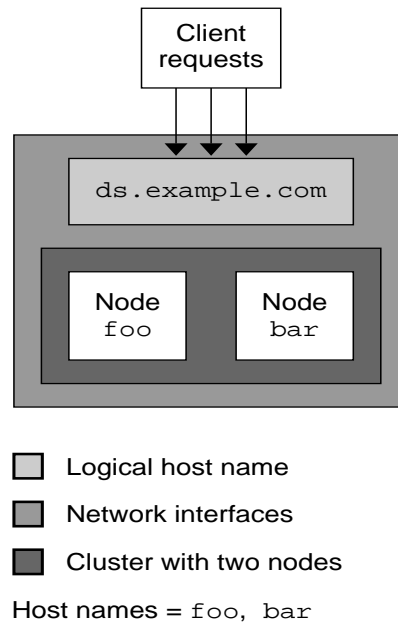
When installing the Sun Cluster HA for Directory Server data service, you configure Directory Server and Administration Server to listen on the logical host name interface so they are not tied to any particular node in the cluster, and the Sun Cluster software can manage failover. In Figure A-1, the nodes are named foo and bar. The logical host name you use during installation as shown in Figure A-1 however would be ds.example.com, not foo or bar. Notice that the logical host name used is a fully qualified domain name.

Refer to the Sun Cluster 3.1 product documentation for more information on these key concepts and for instructions on setting up a logical host name.

Sun Cluster software can also manage failover for the file system resources. When you use HAStoragePlus with the Sun Cluster HA for Directory Server data service, you enable this capability. If you have used HAStorage previously, and intend to migrate from HAStorage with a global file system to HAStoragePlus and a higher performance failover file system, follow the links in Chapter 3, "Finding Patch Update Instructions" on page 37 to find upgrade instructions.

After setting up a logical host name and shared file system, perform the following steps:

1.   Become super user on a node in the cluster.

2. Verify that all network addresses you use have been added to the name service database.

   To avoid failures during name service lookup, ensure as well that the logical host name, the fully qualified domain name, the fully qualified logical host name, and all shared IP addresses are present in the /etc/hosts file on each cluster node. For example, /etc/hosts might contain the following line:

   ```
   192.168.0.99    ds    ds.example.com
   ```

   Also configure name service mapping in /etc/nsswitch.conf on each cluster node to check local files first before trying to access other name services.

3. Create a failover resource group to hold network and application resources. For example:

   ```
   # scrgadm -a -g resource-group [-h node-list]
   ```

   Here *resource-group* specifies the name of the group.

   The optional *node-list* is a comma-separated list of physical node names or IDs identifying potential master nodes for the cluster. The order of the node names determines the order in which the nodes are considered as primary during failover. If all nodes in the cluster are potential masters, it is not necessary to specify the *node-list*.

4. Add logical host name resources to the resource group.

   ```
   # scrgadm -a -L -g resource-group -l logical-host-name [-n netif-list]
   ```

   Here, the optional *netif-list* is a comma-separated list identifying the NAFO groups on each node. If you do not specify this option, scrgadm(1M) attempts to discover a network adapter on the subnet used by each logical host name specified on each node in *node-list* specified in Step 3.

5. Register the resource type for HAStoragePlus.

   ```
   # scrgadm -a -t SUNW.HAStoragePlus
   ```

   Both server data services depend on HAStoragePlus to access data on the shared file system. For more information about HAStoragePlus, follow the links in Chapter 3, "Finding Patch Update Instructions" on page 37.

6. Add the HAStoragePlus resource to the failover resource group created when "Setting Up Network and File System Resources" on page 59.

```
# scrgadm -a -j HAStoragePlus-resource-name -g resource-group \
-t SUNW.HAStoragePlus -x FilesystemMountPoints=mount-point \
-x AffinityOn=TRUE
```

Here, you provide a new *HAStoragePlus-resource-name* to identify the resource. *mount-point* specifies the file system mount point for the *ServerRoot* directory as shown for example in the output of the df(1) command.

7. Enable the resource group and bring it online.

```
# scswitch -Z -g resource-group
```

With the resource group online, you may install the servers.

Installing the Servers

In Sun Cluster HA for Directory Server, both Directory Server and Administration Server run under the control of Sun Cluster. This means that instead of supplying the servers with a fully qualified domain name for the physical node during installation, you provide a fully qualified logical host name that can fail over to a different node.

You perform installation starting with the node online for the logical host name used by directory client applications, then repeating the process for all other cluster nodes that you want to master the Directory Server data service.

NOTE	You install product packages on a node's *local* file system so that each node can be patched separately, but you place directory data on the *shared* cluster file system so that data does not depend on a particular node. This means you can, for example, patch an idle node while another node is providing the directory service.

Installing on the Active Node

For the cluster node that is online for the logical host name used by directory client applications:

1. Install the Solaris packages for both Directory Server and Administration Server on the active node's *local* file system.

   Refer to the links in Chapter 1, "Finding Installation and Upgrade Instructions" on page 15 to determine how to find and install the packages *without configuring them*.

2. Make sure the current node is the active node:

   # scswitch -z -g *resource-group* -h *current-node*

3. Configure Directory Server.

   # /usr/sbin/directoryserver -u 5.2 configure

   When performing this step:

   ○ Place the Directory Server instance, which includes Directory Server data, on the *shared* cluster file system.

   ○ Use the logical host name, *not* the node name.

4. Configure Administration Server, using the same logical host name used to configure Directory Server.

   # /usr/sbin/mpsadmserver configure

5. Stop the servers.

   # /usr/sbin/directoryserver -u 5.2 stop
   # /usr/sbin/mpsadmserver stop

6. When using Directory Server in secure mode only, create an empty file named *ServerRoot*/slapd-*serverID*/keypass to indicate to the cluster that the Directory Server instance runs in secure mode.

   Also create a *ServerRoot*/alias/slapd-*serverID*-pin.txt file, containing the password required to start the instance automatically in secure mode. This allows the cluster to restart the data service without human intervention.

Installing on Other Nodes

Repeat Step 2 on page 63 through Step 4 on page 63 for *each* node you want to master the Directory Server data service using exactly the same configuration data as you used for the first node.

NOTE	Do not remove or relocate any files placed on the shared file system.

Installing the Data Service Packages

The data service packages, `SUNWdsha` and `SUNWasha`, on the product media under `Solaris_`*arch*`/Product/sun_cluster_agents/Solaris_`*version*`/Packages/`, provide the management interfaces for administering the servers as a data services within the cluster.

- On each cluster node that you want to support the Directory Server data service, use the `pkgadd`(1M) utility to install the data service packages.

  ```
  # pkgadd -d dirContainingPackages SUNWasha SUNWdsha
  ```

Configuring the Data Service

Perform the following steps *only* on the cluster node that is online for the logical host name in use by Directory Server:

1. Become super user.

2. Stop Directory Server and Administration Server from the active node if the servers are not already stopped.

   ```
   # /usr/sbin/directoryserver -u 5.2 stop
   # /usr/sbin/mpsadmserver stop
   ```

3. Register the resource types for both data services.

   ```
   # scrgadm -a -t SUNW.dsldap
   # scrgadm -a -t SUNW.mps
   ```

   Here `SUNW.dsldap` and `SUNW.mps` are the predefined resource type names for the data services. `SUNW.dsldap` and `SUNW.mps` define the data services.

4. Add the server resources to the failover resource group created when "Setting Up Network and File System Resources" on page 59.

   ```
   # scrgadm -a -j resource-name-ds -g resource-group -t SUNW.dsldap \
   -y Network_resources_used=logical-host-name \
   -y Port_list=port-number/tcp \
   -x Confdir_list=ServerRoot/slapd-serverID \
   -y Resource_dependencies=HAStoragePlus-resource-name
   ```

```
# scrgadm -a -j resource-name-as -g resource-group -t SUNW.mps \
-y Network_resources_used=logical-host-name \
-y Port_list=port-number/tcp \
-x Confdir_list=ServerRoot \
-y Resource_dependencies=HAStoragePlus-resource-name
```

Here you provide a new *resource-name-ds* to identify the Directory Server instance, a new *resource-name-as* to identify the Administration Server instance, and a new HAStoragePlus resource name.

The *resource-group* parameter is the name of the group specified in "Setting Up Network and File System Resources" on page 59.

The *logical-host-name* identifies the logical host name used for the current Directory Server instance.

The *port-number* is the numbers of the ports on which the server instances listen for client requests, specified in "Installing the Servers" on page 62. Notice the Port_list parameter of each command takes only one entry.

ServerRoot and *ServerRoot*/slapd-*serverID* are paths specified in "Installing the Servers" on page 62. Notice the Confdir_list parameter of each command takes only one entry.

5. Enable the server resources and monitors.

```
# scswitch -e -j resource-name-ds
# scswitch -e -j resource-name-as
```

Here *resource-name-ds* and *resource-name-as* are the names you provided to identify the servers in Step 4.

Example Registration and Configuration

Code Example A-1 shows how you might register and configure the data service for the cluster illustrated in Figure A-1 on page 60.

Code Example A-1   Registering and Configuring the Data Service

```
(Create a failover resource group on the node that is online.)
# scrgadm -a -g ds-resource-group-1 -h foo,bar

(Add a logical hostname resource to the resource group.)
# scrgadm -a -L -g ds-resource-group-1 -l ds

(Register the SUNW.HAStoragePlus resource type.)
# scrgadm -a -t SUNW.HAStoragePlus
```

Code Example A-1   Registering and Configuring the Data Service *(Continued)*

```
(Add the HAStoragePlus resource to the resource group.)
# scrgadm -a -j hasp-resource -g ds-resource-group-1 -t SUNW.HAStoragePlus \
-x FilesystemMountPoints=/shared/ds -x AffinityON=TRUE
# scswitch -e -j hasp-resource

(Bring the resource group online.)
# scswitch -Z -g ds-resource-group-1

(Install packages on each node in the cluster.)

(Stop the servers on the node that is online.)
# /usr/sbin/directoryserver -u 5.2 stop
# /usr/sbin/mpsadmserver stop

(Register the SUNW.dsldap and SUNW.mps resource types.)
# scrgadm -a -t SUNW.dsldap
# scrgadm -a -t SUNW.mps

(Create resources for the servers and add them to the resource group.)
# scrgadm -a -j ds-1 -g ds-resource-group-1 \
-t SUNW.dsldap -y Network_resources_used=ds \
-y Port_list=389/tcp -y Resource_dependencies=hasp-resource \
-x Confdir_list=/shared/ds/slapd-ds-1
# scrgadm -a -j as-1 -g ds-resource-group-1 \
-t SUNW.mps -y Network_resources_used=ds \
-y Port_list=5201/tcp -y Resource_dependencies=hasp-resource \
-x Confdir_list=/shared/ds

(Enable the application resources.)
# scswitch -e -j ds-1
# scswitch -e -j as-1
```

Configuring Extension Properties

Optionally configurable extension properties allow you to configure how the
cluster software handles the application software. For example, you can adjust how
the cluster determines when the data service must fail over.

What You Can Configure

You can configure both standard properties and extension properties. This section
covers extension properties specific to management of Directory Server.

You typically configure resource extension properties using the Cluster Module of the Sun Management Center, or using the scrgadm utility. You can change the extension properties listed in Table A-2 using the scrgadm utility with the -x *parameter*=*value* option.

Table A-2 SUNW.dsldap Resource Extension Properties

Property	Description	Default	Range
Monitor_retry_count	Integer value indicating the number of times the process monitor facility (PMF) restarts the fault monitor during the time window specified by the value of Monitor_retry_interval	4 attempts	-1 to 2,147,483,641 attempts -1 means retry forever.
Monitor_retry_interval	Integer value indicating the time in minutes over which failures of the fault monitor are counted If the number of times the fault monitor fails exceeds the value specified in Monitor_retry_count within this period, the PMF cannot restart the fault monitor.	2 minutes	-1 to 2,147,483,641 minutes -1 specifies an infinite retry interval.
Probe_timeout	Integer value indicating the timeout value in seconds that the fault monitor uses to probe a Directory Server instance	30 seconds	0 to 2,147,483,641 seconds

Refer to the Sun Cluster 3.1 product documentation for more information on Sun Cluster properties.

How the Fault Monitor Operates

The cluster software determines whether the data service is healthy using a fault monitor. The fault monitor probes the data service, and then determines whether the service is healthy or must be restarted based on the results of the probe.

Table A-3   How the Fault Monitor Interprets Probes

Directory Server running in...	Probe Used	Algorithm
Normal mode	ldapsearch	1. Attempt a search. 2. If the search operation results in: • LDAP_SUCCESS, then the service is considered healthy. • An LDAP error, then based on failure history the service is either restarted or fails over to another node. • A problem other than timeout, then the fault monitor probes again depending on Monitor_retry_count and Monitor_retry_interval. • The Probe_timeout duration being exceeded, then the fault monitor probes again depending on Monitor_retry_count and Monitor_retry_interval. Potential causes of timeout include heavy loads on the system, network, or Directory Server instance. Timeout may also indicate that the Probe_timeout value is set too low for the number of Directory Server instances monitored.
Secure mode (SSL)	TCP connect	1. Attempt to connect. 2. If the connection operation: • Succeeds, then the service is considered healthy. • Fails, then based on failure history the service is either restarted or fails over to another node. • Exceeds Probe_timeout, then the service must be restarted.

The fault monitor uses the IP addresses and port numbers you specified when "Configuring the Data Service" on page 64 to carry out probe operations. If Directory Server is configured to listen on two ports, one for SSL traffic and one for normal traffic, the fault monitor probes both ports using TCP connect, following the fault monitoring algorithm used for secure mode ports.

Creating an Additional Directory Server Instance

Perform the following steps:

1. Create an additional Directory Server instance using the Sun Java System Server Console.

   Refer to the *Administration Server Administration Guide* for instructions.

2. Stop the new Directory Server instance on the node that is online for the logical host name in use by the data service.

   ```
   # /usr/sbin/directoryserver -u 5.2 -server serverID stop
   ```

3. Add the Directory Server instance to the failover resource group created in "Setting Up Network and File System Resources" on page 59.

   ```
   # scrgadm -a -j new-ds-resource -g resource-group -t SUNW.dsldap \
   -y Network_resources_used=logical-host-name \
   -y Port_list=port-number/tcp -x Confdir_list=ServerRoot/slapd-serverID \
   -y Resource_dependencies=HAStoragePlus-resource-name
   ```

   Here you provide a *new-ds-resource* to identify the Directory Server instance.

   The *resource-group* parameter is the name of the group specified in "Setting Up Network and File System Resources" on page 59.

   The *logical-host-name* identifies the logical host name used for the instance.

   The *port-number* is the number of the port on which the instance listens for client requests, specified in "Installing the Servers" on page 62. Notice the Port_list parameter takes only one entry.

   ServerRoot and *ServerRoot*/slapd-*serverID* are paths specified in "Installing the Servers" on page 62. Notice the Confdir_list parameter takes only one entry.

   HAStoragePlus-resource-name is the name of the HAStoragePlus resource on which Directory Server depends.

4. Enable the server resources and monitors.

   ```
   # scswitch -e -j new-ds-resource
   ```

   Here *new-ds-resource* is the name you provided to identify the Directory Server in Step 3.

Uninstalling

To remove Sun Cluster HA for Directory Server and the associated Administration Server from the cluster, perform the following steps:

1. Stop the server instances. With a single Directory Server instance for example:

    ```
    # scswitch -n -j resource-name-ds
    # scswitch -n -j resource-name-as
    ```

2. Remove the resources. For example:

    ```
    # scrgadm -r -j resource-name-ds
    # scrgadm -r -j resource-name-as
    ```

3. Remove the resource types from the cluster database.

    ```
    # scrgadm -r -t SUNW.dsldap
    # scrgadm -r -t SUNW.mps
    ```

4. Delete the server configurations.

    a. For each node, unconfigure the Administration Server:

    ```
    # /usr/sbin/mpsadmserver unconfigure
    ```

    b. As Directory Server data and configuration are stored on the shared file system, run the following on any of the nodes where it is configured:

    ```
    # /usr/sbin/directoryserver -u 5.2 unconfigure
    ```

Remove the packages installed, including SUNWdsha and SUNWasha, from each node using the pkgrm(1M) utility.

Glossary

Refer to the *Java Enterprise System Glossary* (`http://docs.sun.com/doc/816-6873`) for a complete list of terms that are used in this documentation set.

Index