Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide

Sun Microsystems, Inc.

Contents

Tables

Figures

Preface

The *Installation Guide* provides step-by-step instructions for installing Directory Service Control Center, Directory Proxy Server, Directory Server, Directory Server Resource Kit, and Identity Synchronization for Windows components of Directory Server Enterprise Edition.

Who Should Use This Book

If you are installing Directory Server Enterprise Edition software for evaluation purposes only, put this guide aside for now, and instead see the quick install instructions in *Sun Java System Directory Server Enterprise Edition 6.0 Evaluation Guide*.

This *Installation Guide* is for administrators deploying Directory Proxy Server, Directory Server, Directory Server Resource Kit, Directory Service Control Center, and Identity Synchronization for Windows software. This document also covers configuration of Identity Synchronization for Windows.

This guide does not cover installation with other Java Enterprise System products. If you plan to install Directory Server and Directory Service Control Center software with other Java ES software, read instead the installation instructions for Java Enterprise System software at http://docs.sun.com/coll/1286.2.

This guide does not cover installation of Directory Editor software. If you plan to install Directory Editor software, read instead the installation instructions in the *Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide*. Make sure you read Chapter 6, "Directory Editor Bugs Fixed and Known Problems," in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* before you install Directory Editor.

TABLE P–1 Where to Find Installation Instructions

Directory Server Enterprise Edition Component	Where to Find Installation Instructions
Directory Service Control Center	Chapter 1
Directory Server	Chapter 1
Directory Proxy Server	Chapter 1

TABLE P–1 Where to Find Installation Instructions *(Continued)*

Directory Server Enterprise Edition Component	Where to Find Installation Instructions
Identity Synchronization for Windows	Part II
Directory Editor	First read "Known Problems and Limitations in Directory Editor" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*. Then read *Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide*
Directory Server Resource Kit	Chapter 1

Before You Read This Book

Review pertinent information in the *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*.

If you are deploying Directory Server Enterprise Edition software in production, also review pertinent information in the *Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide*.

Readers installing Identity Synchronization for Windows should be familiar with the following technologies:

- Directory Server
- Microsoft Active Directory or Windows NT authentication
- Lightweight Directory Access Protocol (LDAP)
- Java technology
- Extensible Markup Language (XML)
- Public-key cryptography and Secure Sockets Layer (SSL) protocol
- Intranet, extranet, and Internet security
- The role of digital certificates in an enterprise

How This Book Is Organized

Part I covers installation of Directory Proxy Server, Directory Server, Directory Server Resource Kit, and Directory Service Control Center on supported systems.

Part II covers installation of Identity Synchronization for Windows on supported systems.

Directory Server Enterprise Edition Documentation Set

This Directory Server Enterprise Edition documentation set explains how to use Sun Java System Directory Server Enterprise Edition to evaluate, design, deploy, and administer directory services. In addition, it shows how to develop client applications for Directory Server Enterprise Edition. The Directory Server Enterprise Edition documentation set is available at http://docs.sun.com/coll/1224.1.

For an introduction to Directory Server Enterprise Edition, review the following documents in the order in which they are listed.

TABLE P–2 Directory Server Enterprise Edition Documentation

Document Title	Contents
Sun Java System Directory Server Enterprise Edition 6.0 Release Notes	Contains the latest information about Directory Server Enterprise Edition, including known problems.
Sun Java System Directory Server Enterprise Edition 6.0 Documentation Center	Contains links to key areas of the documentation set.
Sun Java System Directory Server Enterprise Edition 6.0 Evaluation Guide	Introduces the key features of this release. Demonstrates how these features work and what they offer in the context of a fictional deployment that you can implement on a single system.
Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide	Explains how to plan and design highly available, highly scalable directory services based on Directory Server Enterprise Edition. Presents the basic concepts and principles of deployment planning and design. Discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Directory Server Enterprise Edition.
Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide	Explains how to install the Directory Server Enterprise Edition software. Shows how to select which components to install, configure those components after installation, and verify that the configured components function properly. For instructions on installing Directory Editor, go to http://docs.sun.com/coll/DirEdit_05q1. Make sure you read the information in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* concerning Directory Editor before you install Directory Editor.
Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide	Provides instructions for upgrading components from earlier versions of Directory Server, Directory Proxy Server, and Identity Synchronization for Windows.

TABLE P–2   Directory Server Enterprise Edition Documentation   *(Continued)*

Document Title	Contents
Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide	Provides command-line instructions for administering Directory Server Enterprise Edition.
	For hints and instructions on using the Directory Service Control Center, DSCC, to administer Directory Server Enterprise Edition, see the online help provided in DSCC.
	For instructions on administering Directory Editor, go to http://docs.sun.com/coll/DirEdit_05q1.
	For instructions on installing and configuring Identity Synchronization for Windows, see Part II.
Sun Java System Directory Server Enterprise Edition 6.0 Developer's Guide	Shows how to develop server plug-ins with the APIs that are provided as part of Directory Server Enterprise Edition.
Sun Java System Directory Server Enterprise Edition 6.0 Reference	Introduces the technical and conceptual foundations of Directory Server Enterprise Edition. Describes its components, architecture, processes, and features. Also provides a reference to the developer APIs.
Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference	Describes the command-line tools, schema objects, and other public interfaces that are available through Directory Server Enterprise Edition. Individual sections of this document can be installed as online manual pages.
Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide	Provides general guidelines and best practices for planning and deploying Identity Synchronization for Windows

Related Reading

The SLAMD Distributed Load Generation Engine (SLAMD) is a Java™ application that is designed to stress test and analyze the performance of network-based applications. It was originally developed by Sun Microsystems, Inc. to benchmark and analyze the performance of LDAP directory servers. SLAMD is available as an open source application under the Sun Public License, an OSI-approved open source license. To obtain information about SLAMD, go to http://www.slamd.com/. SLAMD is also available as a java.net project. See https://slamd.dev.java.net/.

Java Naming and Directory Interface (JNDI) technology supports accessing the Directory Server using LDAP and DSML v2 from Java applications. For information about JNDI, see http://java.sun.com/products/jndi/. The *JNDI Tutorial* contains detailed descriptions and examples of how to use JNDI. This tutorial is at http://java.sun.com/products/jndi/tutorial/.

Directory Server Enterprise Edition can be licensed as a standalone product, as a component of Sun Java Enterprise System, as part of a suite of Sun products, such as the Sun Java Identity Management Suite, or as an add-on package to other software products from Sun. Java

Enterprise System is a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If Directory Server Enterprise Edition was licensed as a component of Java Enterprise System, you should be familiar with the system documentation at `http://docs.sun.com/coll/1286.2`.

Identity Synchronization for Windows uses Message Queue with a restricted license. Message Queue documentation is available at `http://docs.sun.com/coll/1307.2`.

Identity Synchronization for Windows works with Microsoft Windows password policies.

- Information about password policies for Windows 2003 is available in the Microsoft documentation online.

- Information about changing passwords, and about group policies in Windows 2003 is available the Microsoft documentation online.

- Information about the Microsoft Certificate Services Enterprise Root certificate authority is available in the Microsoft support documentation online.

- Information about configuring LDAP over SSL on Microsoft systems is available in the Microsoft support documentation online.

Redistributable Files

Directory Server Enterprise Edition does not provide any files that you can redistribute.

Default Paths and Command Locations

This section explains the default paths used in the documentation, and gives the locations of commands on different operating systems and deployment types.

Default Paths

The table in this section describes the default paths that are used in this document. For full descriptions of the files installed, see also Chapter 15, "Directory Server File Reference," in *Sun Java System Directory Server Enterprise Edition 6.0 Reference*, Chapter 26, "Directory Proxy Server File Reference," in *Sun Java System Directory Server Enterprise Edition 6.0 Reference*, or Appendix A, "Directory Server Resource Kit File Reference," in *Sun Java System Directory Server Enterprise Edition 6.0 Reference*.

TABLE P–3   Default Paths

Placeholder	Description	Default Value
install-path	Represents the base installation directory for Directory Server Enterprise Edition software. The software is installed in directories below this base *install-path*. For example, Directory Server software is installed in *install-path*/ds6/.	When you install from a zip distribution using dsee_deploy(1M), the default *install-path* is the current directory. You can set the *install-path* using the -i option of the dsee_deploy command. When you install from a native package distribution, such as you would using the Java Enterprise System installer, the default *install-path* is one of the following locations: ■ Solaris systems - /opt/SUNWdsee/. ■ HP-UX systems - /opt/sun/. ■ Red Hat systems - /opt/sun/. ■ Windows systems - C:\Program Files\Sun\JavaES5\DSEE.
instance-path	Represents the full path to an instance of Directory Server or Directory Proxy Server. The documentation uses /local/ds/ for Directory Server and /local/dps/ for Directory Proxy Server.	No default path exists. Instance paths must nevertheless always be found on a *local* file system. The following directories are recommended: /var on Solaris systems /global if you are using Sun Cluster
serverroot	Represents the parent directory of the Identity Synchronization for Windows installation location	Depends on your installation. Note the concept of a *serverroot* no longer exists for Directory Server.
isw-*hostname*	Represents the Identity Synchronization for Windows instance directory	Depends on your installation
/path/to/cert8.db	Represents the default path and file name of the client's certificate database for Identity Synchronization for Windows	*current-working-dir*/cert8.db
serverroot/isw-*hostname*/logs/	Represents the default path to the Identity Synchronization for Windows local logs for the System Manager, each connector, and the Central Logger	Depends on your installation
serverroot/isw-*hostname*/logs/central/	Represents the default path to the Identity Synchronization for Windows central logs	Depends on your installation

Command Locations

The table in this section provides locations for commands that are used in Directory Server Enterprise Edition documentation. To learn more about each of the commands, see the relevant man pages.

TABLE P–4 Command Locations

Command	Java ES, Native Package Distribution	Zip Distribution
cacaoadm	Solaris - `/usr/sbin/cacaoadm`	Solaris - *install-path*/dsee6/ `cacao_2.0/usr/lib/cacao/bin/cacaoadm`
	Red Hat, HP-UX - `/opt/sun/cacao/bin/cacaoadm`	Red Hat, HP-UX - *install-path*/dsee6/ `cacao_2.0/cacao/bin/cacaoadm`
	Windows - *install-path*\share\ `cacao_2.0\bin\cacaoadm.bat`	Windows - *install-path*\ `dsee6\cacao_2.0\bin\cacaoadm.bat`
certutil	Solaris - `/usr/sfw/bin/certutil`	*install-path*/dsee6/bin/certutil
	Red Hat, HP-UX - `/opt/sun/private/bin/certutil`	
dpadm(1M)	*install-path*/dps6/bin/dpadm	*install-path*/dps6/bin/dpadm
dpconf(1M)	*install-path*/dps6/bin/dpconf	*install-path*/dps6/bin/dpconf
dsadm(1M)	*install-path*/ds6/bin/dsadm	*install-path*/ds6/bin/dsadm
dsccmon(1M)	*install-path*/dscc6/bin/dsccmon	*install-path*/dscc6/bin/dsccmon
dsccreg(1M)	*install-path*/dscc6/bin/dsccreg	*install-path*/dscc6/bin/dsccreg
dsccsetup(1M)	*install-path*/dscc6/bin/dsccsetup	*install-path*/dscc6/bin/dsccsetup
dsconf(1M)	*install-path*/ds6/bin/dsconf	*install-path*/ds6/bin/dsconf
dsee_deploy(1M)	Not provided	*install-path*/dsee6/bin/dsee_deploy
dsmig(1M)	*install-path*/ds6/bin/dsmig	*install-path*/ds6/bin/dsmig
entrycmp(1)	*install-path*/ds6/bin/entrycmp	*install-path*/ds6/bin/entrycmp
fildif(1)	*install-path*/ds6/bin/fildif	*install-path*/ds6/bin/fildif
idsktune(1M)	*install-path*/dsrk6/bin/idsktune	*install-path*/dsrk6/bin/idsktune

TABLE P–4 Command Locations   *(Continued)*

Command	Java ES, Native Package Distribution	Zip Distribution
insync(1)	*install-path*/ds6/bin/insync	*install-path*/ds6/bin/insync
ns-accountstatus(1M)	*install-path*/ds6/bin/ns-accountstatus	*install-path*/ds6/bin/ns-accountstatus
ns-activate(1M)	*install-path*/ds6/bin/ns-activate	*install-path*/ds6/bin/ns-activate
ns-inactivate(1M)	*install-path*/ds6/bin/ns-inactivate	*install-path*/ds6/bin/ns-inactivate
repldisc(1)	*install-path*/ds6/bin/repldisc	*install-path*/ds6/bin/repldisc
schema_push(1M)	*install-path*/ds6/bin/schema_push	*install-path*/ds6/bin/schema_push
smcwebserver	Solaris, Linux, HP-UX - /usr/sbin/smcwebserver Windows - *install-path*\share\ webconsole\bin\smcwebserver	This command pertains only to Directory Service Control Center, which is not available in the zip distribution.
wcadmin	Solaris, Linux, HP-UX - /usr/sbin/wcadmin Windows - *install-path*\share\ webconsole\bin\wcadmin	This command pertains only to Directory Service Control Center, which is not available in the zip distribution.

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–5   Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your .login file. Use ls -a to list all files. machine_name% you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	machine_name% **su** Password:
AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is rm *filename*.

TABLE P–5 Typographic Conventions   *(Continued)*

Typeface	Meaning	Example
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Shell Prompts in Command Examples

The following table shows default system prompts and superuser prompts.

TABLE P–6 Shell Prompts

Shell	Prompt
C shell on UNIX and Linux systems	`machine_name%`
C shell superuser on UNIX and Linux systems	`machine_name#`
Bourne shell and Korn shell on UNIX and Linux systems	`$`
Bourne shell and Korn shell superuser on UNIX and Linux systems	`#`
Microsoft Windows command line	`C:\`

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–7 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and command options.	`ls [-l]`	The `-l` option is not required.
`{	}`	Contains a set of choices for a required command option.	`-d {y|n}`	The `-d` option requires that you use either the y argument or the n argument.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
`-`	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.

TABLE P–7	Symbol Conventions	*(Continued)*	
Symbol	**Description**	**Example**	**Meaning**
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (http://www.sun.com/documentation/)
- Support (http://www.sun.com/support/)
- Training (http://www.sun.com/training/)

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – 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.

Searching Sun Product Documentation

Besides searching for Sun product documentation from the docs.sun.com web site, you can use a search engine of your choice by typing the following syntax in the search field:

search-term site:docs.sun.com

For example, to search for Directory Server, type the following:

"Directory Server" site:docs.sun.com

To include other Sun web sites in your search, such as java.sun.com, www.sun.com, and developers.sun.com, use sun.com in place of docs.sun.com in the search field.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-0993.

P A R T I

Installing Directory Service Control Center, Directory Proxy Server, Directory Server, and Directory Server Resource Kit

This part includes the following chapters.

- Chapter 1 explains how to install Directory Service Control Center, Directory Proxy Server, and Directory Server, Directory Server Resource Kit on supported systems.

- Chapter 2 explains how to remove Directory Proxy Server, Directory Server, Directory Server Resource Kit, and Directory Service Control Center.

For help with installation of Identity Synchronization for Windows software, see Part II.

This guide does not cover installation with other Java Enterprise System products. If you plan to install Directory Server and Directory Service Control Center software with other Java ES software, read instead the installation instructions for Java Enterprise System software at `http://docs.sun.com/coll/1286.2`.

This guide does not cover installation of Directory Editor software. If you plan to install Directory Editor software, read instead the installation instructions in the *Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide*. Make sure you read Chapter 6, "Directory Editor Bugs Fixed and Known Problems," in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* before you install Directory Editor.

1

Installation

This chapter guides you in installing Directory Server Enterprise Edition software.

- "Before You Install" on page 33 covers concepts you must understand before installing in a production environment.
- "Directory Server Enterprise Edition Software Distributions" on page 38 contrasts the Directory Server Enterprise Edition software distributions.
- "Installation in Solaris Zones" on page 41 addresses what to consider when installing in a Solaris zone.
- "Software Installation" on page 42 provides step by step instructions on how to install Directory Server Enterprise Edition software.
- "Server Instance Creation" on page 66 provides step by step instructions on how to create server instances after you install the software.
- "Working With Sun Cryptographic Framework on Solaris 10 Systems" on page 76 provides instructions for deployments that use SSL hardware acceleration.

At the end of this chapter, you will have verified that the software that you installed works as expected. You can then continue to configure the software as described in the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

Before You Install

Before installing Directory Server Enterprise Edition software in a production environment, obtain the plans for deployment that were created with the help of *Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide*. With the plans in hand, read this section to gauge how to approach installation for your deployment.

This chapter includes the following sections.

- "The Administration Framework and Installation" on page 34 briefly covers administration framework concepts that are key for installation in a production environment.

- "Comparison of Single System And Distributed Installation" on page 36 compares and contrasts installations that involve a single host system with installations that involve multiple systems.

The Administration Framework and Installation

This section highlights key aspects of the administration framework you must understand before installing server software in a production environment. This section does not address the developer and performance tuning tools provided with Directory Server Resource Kit. You can install such tools independently of the administration framework.

Before you read this section, read "Directory Server Enterprise Edition Administration Model" in *Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide*. In particular, consider the figure in that section which shows the network traffic flows. The figure shows network traffic flows between the configuration management tools, Directory Service Control Center (DSCC), dsconf(1M), and dpconf(1M), the local administration agents, and servers. The figure also shows communication between the local agents, the local command line tools, dsadm(1M) and dpadm(1M), and the servers that you manage.

Notice the command line management and monitoring tools, dsconf(1M) and dpconf(1M), require only LDAP access to the servers that you manage. LDAP traffic typically flows through the default ports, 389 for LDAP and 636 for secure LDAP using SSL. When you create servers as a non-root user, the default ports are 1389 for LDAP, and 1636 for secure LDAP using SSL.

By convention, only root can install software using reserved port numbers less than 1024. Solaris systems allow the administrator to permit non-root users to use privileged ports, using role-based access control (RBAC).

DSCC is a web application. DSCC runs inside the framework known as Sun Java Web Console. You typically install DSCC on only one system in your deployment. You then manage all your servers from that installation of DSCC. You access DSCC through a browser by going to the secure Java Web Console URL, which by default is https://*hostname*:6789.

DSCC requires LDAP access to the servers for online management operations. DSCC also requires Java Management Extension (JMX) access to agents installed alongside the servers. The agents perform server process management operations on behalf of DSCC, operations that cannot be performed through LDAP on a running server. You can then work through a browser to DSCC to create and to start new servers.

As part of the normal installation process, you install the local DSCC agents alongside server software. DSCC contacts the agents over the network using a specific port number. You must therefore either accept the default port number, 11162, or specify a different port number.

The agents run inside a common agent container on the server system. This common agent container provides its agents with a single external port for management applications. The common agent container also consolidates resources to save resources on systems where multiple local agents share the container. The common agent container is in fact the agent that listens for DSCC on the default port number, 11162, routing management traffic to other agents. DSCC thus communicates with local agents through the common agent container. For troubleshooting purposes, a common agent container can be managed independently using the cacaoadm command.
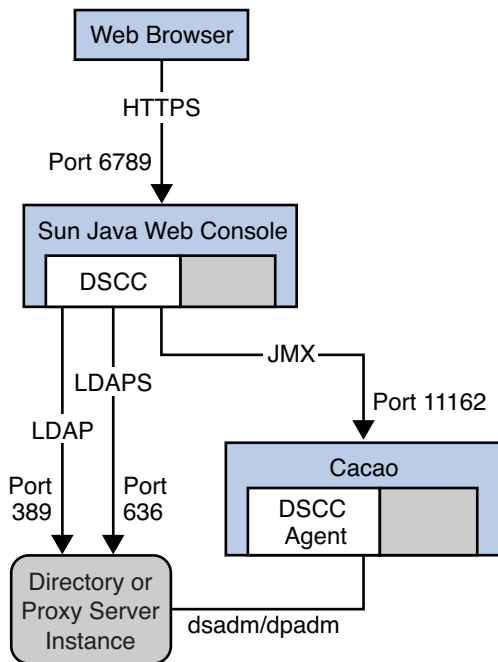
FIGURE 1–1   Ports and Protocols

Each time that you install Directory Server Enterprise Edition software from the zip distribution, you also install an instance of the common agent container. Therefore, when you install multiple versions in parallel on the same host system, *only one version can use the default port*. You can install from the zip distribution where a common agent container instance already uses the default port. You must then specify a different port number for the additional common agent container instance.

Server software installation is a three stage process.

1. Install configuration management software.

    During this stage, configuration management tools are installed. DSCC is also initialized.

    As DSCC stores its configuration data in its own, private Directory Server instance, Directory Server is installed from native packages alongside DSCC.

2. Install server software on the systems where you plan to run server instances.

    During this stage, server software, required libraries, local administration tools, and local agents are installed where server instances run.

    At the close of this stage, no servers are running. Yet, all the software is in place to allow you to set up directory services.

3. Create and configure server instances on the systems.

    During this stage, Directory Server and Directory Proxy Server instances are created. Instances are created either through a web browser with DSCC, or with the local administration tools that are installed alongside the server software. Server instances are then configured either through Directory Service Control Center or through the configuration management command line tools.

The first two stages are combined when you install everything on a single host system. DSCC nevertheless uses the local agents to perform certain operations on the servers. Thus, the local agents must still be installed in a local common agent container.

Comparison of Single System And Distributed Installation

This section compares and contrasts single host system installations with installations that involve multiple systems.

This section explains the outcomes of two basic choices about your installation.

1. The choice whether to install DSCC and configuration management tools on the *same host* as the servers that you manage. Alternatively, you can install the tools on a *different host* from the servers that you manage remotely.

2. The choice whether to create multiple server instances on the *same host*, or create each server instance on a *different host*.

Where You Install Directory Service Control Center

Installing DSCC on the *same host* as the servers that you manage provides a quick and simple solution for evaluation and development. This solution is not recommended for production installations where you rely on redundant systems and on server replica to provide high availability.

When you install DSCC you also install Directory Server software. DSCC uses its own private instance of Directory Server to store configuration information. If you also install the local agent for Directory Server alongside DSCC, you can create Directory Server instances on the system through a web browser to DSCC. You can do so without having to know additional host names and port numbers.

You can install DSCC on a *different host* from the servers you manage remotely. This solution is recommended for production installations where you rely on redundant systems and on server replica to provide high availability.
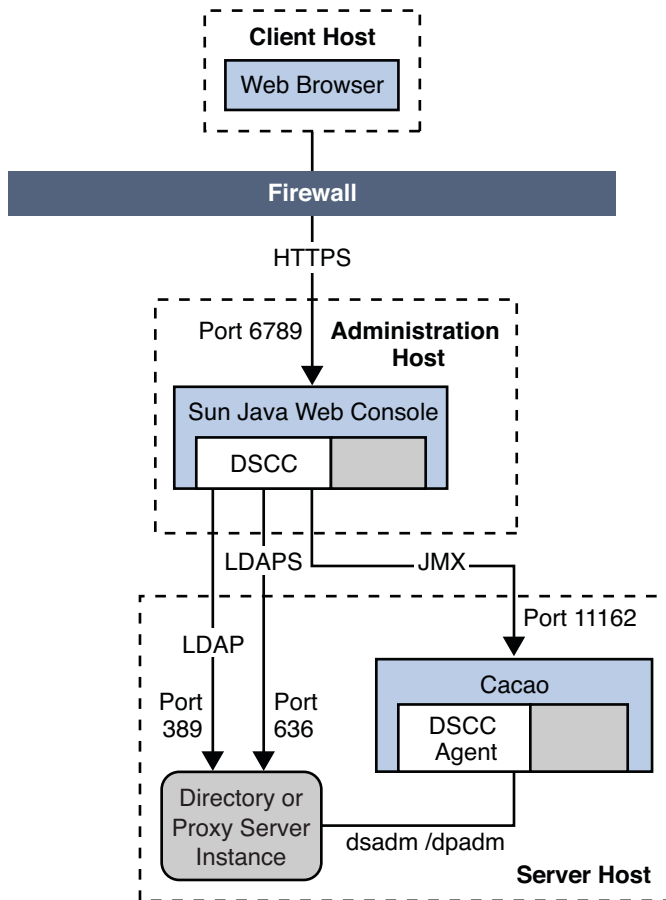
FIGURE 1–2   Administration Host and Server Host on Different Systems

When you install DSCC on the administration host, you must be root. However, you can then use DSCC installed on the administration host to manage server hosts installed as non-root.

For example, you install DSCC on a server or even a suitable workstation outside the data center. You also install server software from the zip distribution on server hosts inside the data center, performing such installations as non-root. Over secure LDAP and JMX, you can then create, configure, and manage all your servers through a web browser to DSCC on the administration host.

Where You Create Server Instances

For production installations, you rely on redundant systems, load balancing, failover capabilities, and server replica to provide high availability. You therefore typically create servers on multiple host systems. Yet, more powerful host systems might each house multiple server instances.

When you create multiple server instances on a single host system, only one server instance can listen on the default ports. As long as you install Directory Server Enterprise Edition software only once, multiple server instances can share the same common agent container.

When you install multiple Directory Server Enterprise Edition versions on a system, each version comes with its own common agent container. Only one of those common agent containers can listen on the default port for JMX management traffic.

Directory Server Enterprise Edition Software Distributions

This section compares the different Directory Server Enterprise Edition software distributions available.

- "Java Enterprise System Distribution" on page 39 introduces the native package distribution that is provided as part of Sun Java Enterprise System. The Java Enterprise System distribution *does* include Directory Service Control Center.

  You install software from the Java Enterprise System distribution with the Java Enterprise System installer.
- "Zip Distribution" on page 39 introduces the zip distribution, which supports non-root installations. The zip distribution *does not* include Directory Service Control Center.
- "Comparison of Distributions" on page 40 summarizes the software that is provided in each distribution.
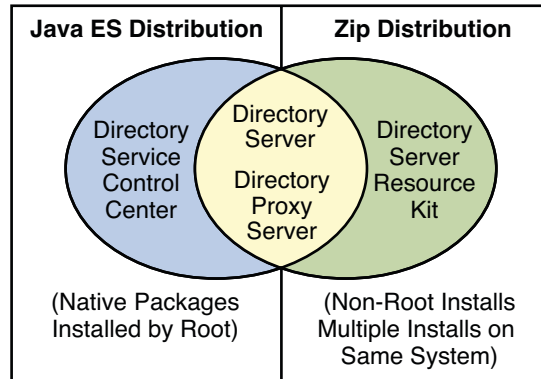
FIGURE 1–3 The Two Software Distributions

Java Enterprise System Distribution

This section introduces the *Java Enterprise System distribution*, which comes with the Java ES installer.

The Java ES installer offers a graphical wizard, a command-line interactive wizard, and also silent installation capabilities to add natively packaged software to your system. As this distribution is based on native packages, you must be root to perform the installation with the Java ES installer.

All Java ES software can work together, relying on a common framework of basic components and of libraries. You can therefore install all of the software products together on a single system if you choose to do so.

The Java ES installation software also facilitates patching, and migration from previous Java ES versions, including upgrade of common components. Furthermore, the software that you install integrates with the system, so you can configure directory services to restart automatically when the operating system reboots. With a native package based installation, you benefit from the package versioning and patching tools that are part of the operating system.

This guide does not describe all installation alternatives available using the Java ES installer. This guide does, however, address the key Java ES installer wizard screens related to Directory Server Enterprise Edition software installation. For detailed instructions on using all features of the Java ES installer, see the Java Enterprise System documentation at http://docs.sun.com/coll/1286.2.

Zip Distribution

This section introduces the *zip distribution*, which provides the dsee_deploy(1M) command-line installer.

This distribution offers self-contained software that you can install anywhere on local disk where you have write permission. You can both install, and also administer zip distribution software as a non-root user.

On Windows systems in order to allow remote management capabilities when installing the zip distribution, you must install either as Administrator, or as a user who belongs directly to the local Administrators group.

As zip distribution software is self-contained, each software installation performed from the zip distribution is independent. You can therefore install software from multiple zip distribution versions on the same system. Your system administrator must manually configure the software that you install to restart when the operating system reboots, however.

Furthermore, with the zip distribution, you must keep careful track of what you have installed, and which patches you have applied.

Comparison of Distributions

This section identifies which software is supported in each distribution.

Both distributions allow you to create and configure Directory Server and Directory Proxy Server instances as non-root.

Directory Server Enterprise Edition Software Component	Java Enterprise System Distribution	Zip Distribution
Directory Service Control Center	Provided	Not provided in this distribution
Directory Server	Provided	Provided, installable with dsee_deploy
Directory Proxy Server	Provided	Provided, installable with dsee_deploy
Directory Editor	Not provided in this distribution	Provided, but *not* installable with dsee_deploy
Identity Synchronization for Windows	Not provided in this distribution	Provided, but *not* installable with dsee_deploy
Directory Server Resource Kit, including Directory SDK for C and Directory SDK for Java	Not provided in this distribution	Provided, installable with dsee_deploy

Identity Synchronization for Windows and Directory Editor software are bundled with the zip distribution, but are not installed using the dsee_deploy command. This guide does cover Identity Synchronization for Windows installation, however. See Part II.

This guide does not cover installation of Directory Editor software. If you plan to install Directory Editor software, then read instead the installation instructions in the *Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide*.

Installation in Solaris Zones

This section addresses what to consider when installing Directory Server Enterprise Edition in a Solaris zone.

Global and full local Solaris zones present Directory Server Enterprise Edition software with complete systems. Directory Server Enterprise Edition software within a full local zone, as within the global zone, runs as if the zone were an independent physical system. The software does not share services or file locations with other zones. In global zones, and in full local zones, you therefore install as if you were on an independent system.

In sparse zones, you can install some services to be used in system-wide fashion. Single instances of Java Enterprise System common component services can therefore be used by multiple Java ES server instances. For example, Directory Server Enterprise Edition software in sparse zones can use the same Common Agent Container and Java ES Monitoring Framework installed in the global zone. You must, however, install the system-wide services before you can complete installation of sparse zone software that depends on the system-wide services.

Directory Server Enterprise Edition does not, however, require you to use system-wide services when you install in a sparse zone. When you install self-contained software from the zip distribution, you also install the common component services in the sparse zone. Therefore, zip distribution installations in sparse zones resemble installations on independent systems.

The following table outlines constraints for Directory Server Enterprise Edition installations, which pertain essentially to installations in sparse zones.

Directory Server Enterprise Edition Software Component	Software Distribution	Constraints Installing in Global or Full Local Zone	Constraints For Sparse Zone Installations
Directory Service Control Center	Java Enterprise System distribution	No constraints	First install Java Enterprise System common components in the global zone, then install Directory Service Control Center in the sparse zone.

Directory Server Enterprise Edition Software Component	Software Distribution	Constraints Installing in Global or Full Local Zone	Constraints For Sparse Zone Installations
Directory Server	Java Enterprise System distribution	No constraints	First install Java Enterprise System common components in the global zone, then install Directory Server in the sparse zone.
	Zip distribution	No constraints	No constraints
Directory Proxy Server	Java Enterprise System distribution	No constraints	First install Java Enterprise System common components in the global zone, then install Directory Proxy Server in the sparse zone.
	Zip distribution	No constraints	No constraints
Directory Editor	Zip distribution	No constraints	The web application container must allow installation in the sparse zone.
Identity Synchronization for Windows	Zip distribution	No constraints	Not supported
Directory Server Resource Kit	Zip distribution	No constraints	No constraints

For details about installation from the Java Enterprise System distribution in sparse zones, see the Java Enterprise System documentation, `http://docs.sun.com/coll/1286.2`.

Software Installation

This section covers basic installation. After you install server software, see "Server Instance Creation" on page 66 for instructions on creating server instances.

- "To Install Directory Service Control Center From Native Packages" on page 43
- "To Troubleshoot Directory Service Control Center Access" on page 50
- "To Install Only Directory Server From Native Packages" on page 51
- "To Install Only Directory Server From the Zip Distribution" on page 54
- "To Install Only Directory Proxy Server From Native Packages" on page 57
- "To Install Only Directory Proxy Server From the Zip Distribution" on page 60
- "To Install Only Directory Server Resource Kit From the Zip Distribution" on page 63
- "Environment Variables" on page 65

▼ To Install Directory Service Control Center From Native Packages

This procedure covers installation of Directory Service Control Center, also known as DSCC, and remote administration command-line tools.

You must be root to perform this procedure.

When you install DSCC, you automatically also install Directory Server from native packages. DSCC uses its own local instance of Directory Server to store information about your directory service configuration. The instance is referred to as the DSCC Registry.

You can use the Directory Server software that is installed alongside DSCC to create your own additional Directory Server instances on the system.

Before You Begin   Obtain the Java Enterprise System distribution for this installation, as shown in the following figure.
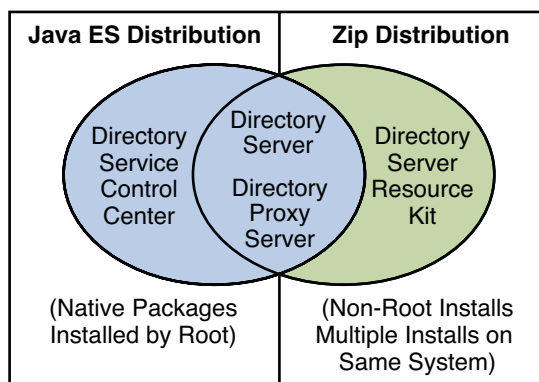
Complete the worksheet that follows for your installation.

Requisite Information	Hints	Your Answers
Hostname of the system where you install DSCC		
root password for the system		
Java Web Console URL	Default: https://localhost:6789	
Directory Service Manager password		

1   **Install prerequisite patches or service packs for your platform.**

    See "Operating System Requirements" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* for details.

2   **With the Java Enterprise System distribution, run the Java ES installer as** root.

    `# ./installer`

3   **Select the Directory Service Control Center component for installation.**
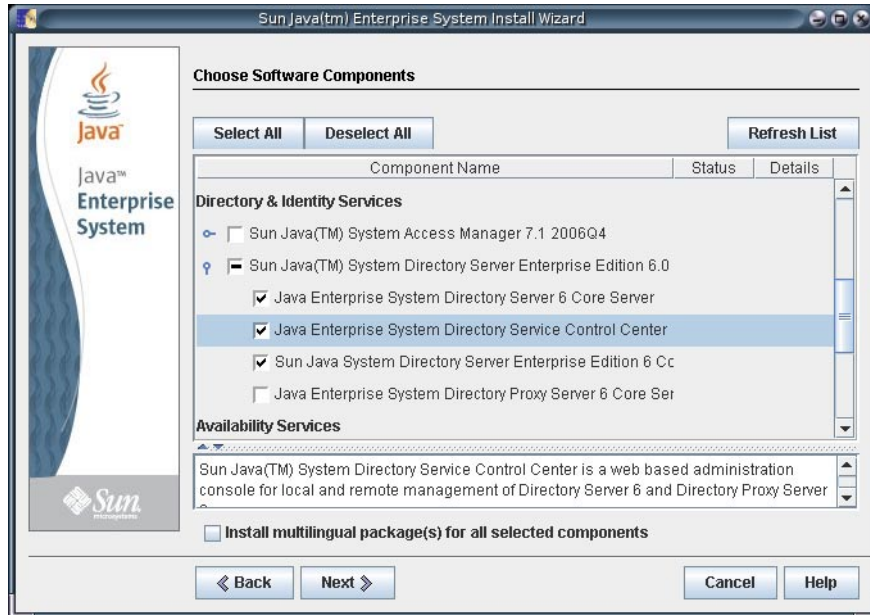
4   **Choose to configure the software later, as you will register the software and create server instances after installation.**
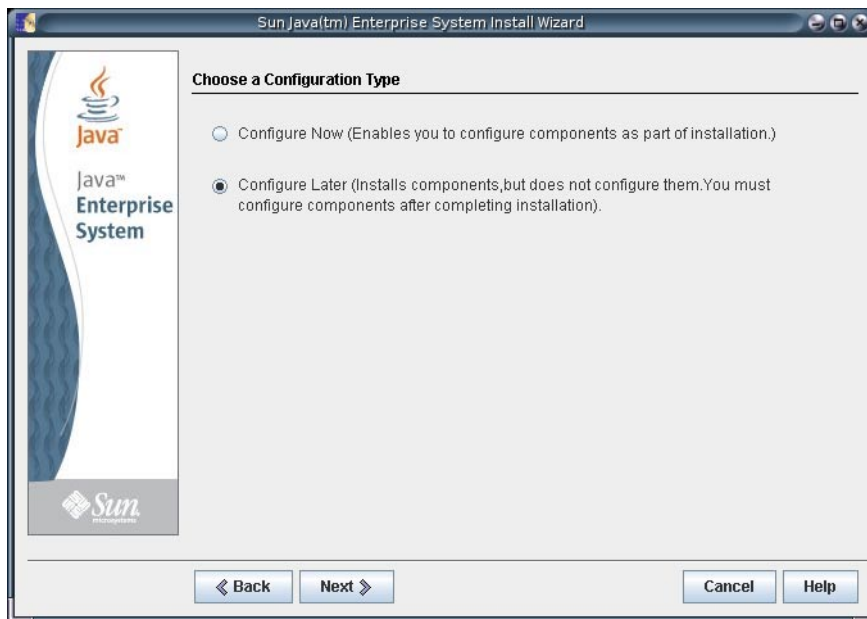
5 Complete installation with the Java ES installer.

After you complete installation, the native packages are installed on the system.

6 Initialize DSCC with the dsccsetup initialize **command.**

For example, on a Solaris system the following command performs initialization.

```
root# /opt/SUNWdsee/dscc6/bin/dsccsetup initialize
***
Registering DSCC Application in Sun Java(TM) Web Console
This operation is going to stop Sun Java(TM) Web Console.
Do you want to continue ? [y,n] y
Stopping Sun Java(TM) Web Console...
Registration is on-going. Please wait...
DSCC is registered in Sun Java(TM) Web Console
Restarting Sun Java(TM) Web Console
Please wait : this may take several seconds...
Sun Java(TM) Web Console restarted successfully
***
Registering DSCC Agent in Cacao...
Checking Cacao status...
Starting Cacao...
DSCC agent has been successfully registered in Cacao.
***
Choose password for Directory Service Manager:
Confirm password for Directory Service Manager:
Creating DSCC registry...
```

```
DSCC Registry has been created successfully
***
```

The dsccsetup command is located in *install-path*/dscc6/bin/dsccsetup. See "Default Paths" on page 23 to determine the default *install-path* for your system.

7   Access DSCC through Java Web Console in your browser.

a.   Login to Java Web Console.

By default, the URL to access Java Web Console is of the following form.

https://*hostname*:6789

b.   Click the Directory Service Control Center link.
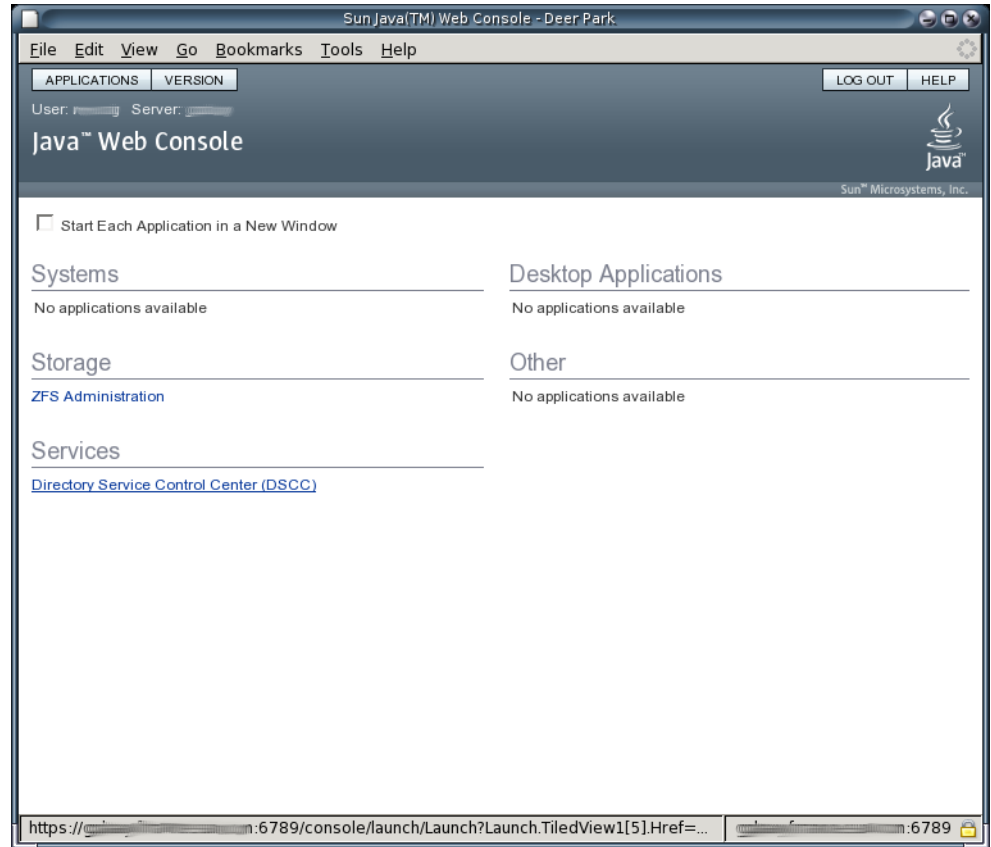
c. **Login as Directory Service Manager.**

Directory Service Manager's entry is stored in the Directory Service Control Center registry. Directory Service Manager has administrator access to Directory Service Control Center. Directory Service Manager also has administrator access to the server instances registered with Directory Service Control Center.
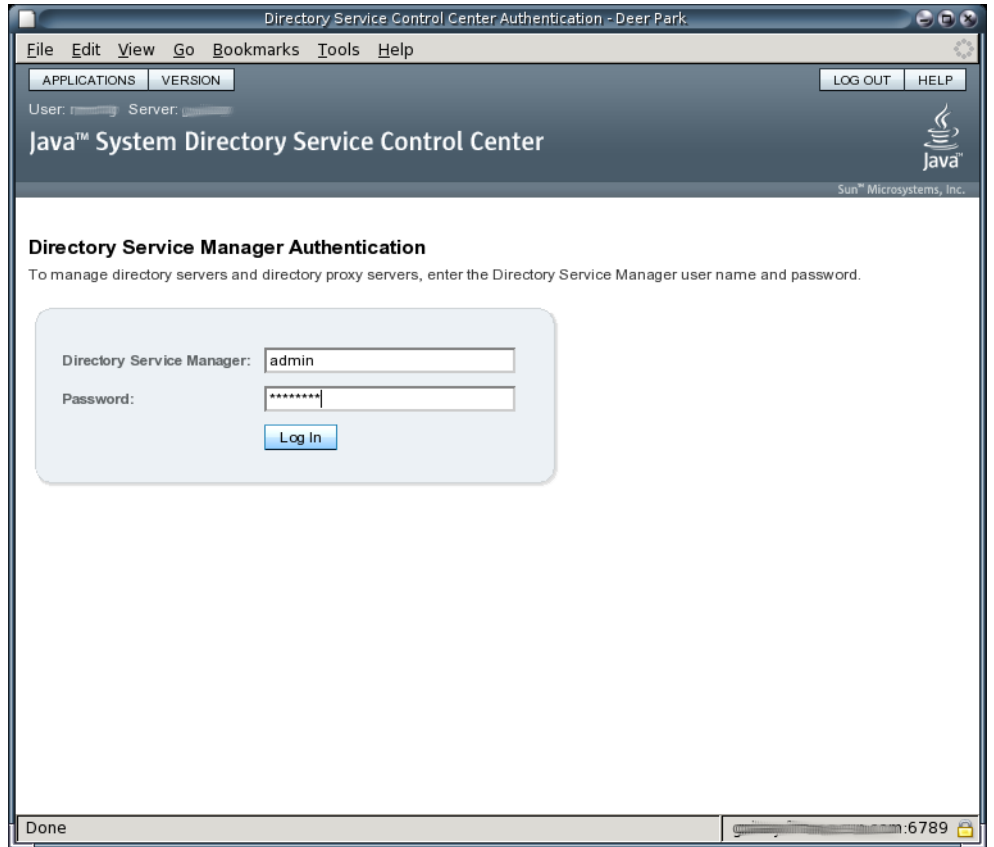
d. **Begin managing your servers through Directory Service Control Center.**
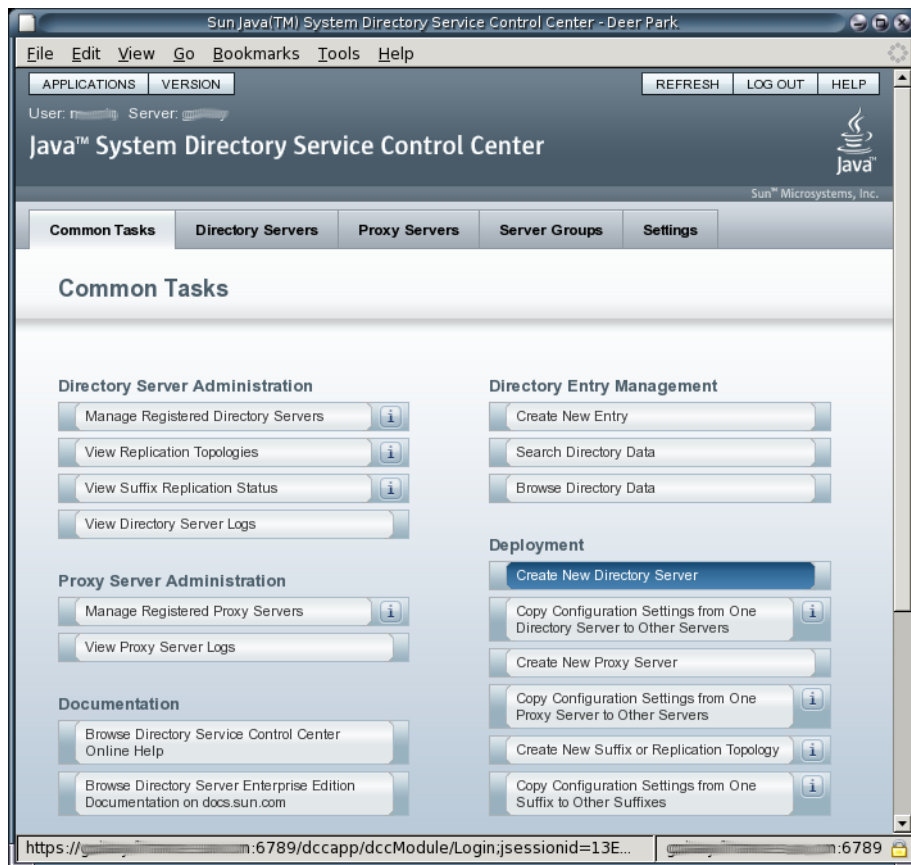
8 **After Directory Service Control Center is working, enable Java Web Console to restart when the system reboots.**

On a Solaris system, the following command enables restart upon reboot.

```
root# /usr/sbin/smcwebserver enable
```

For the exact location of this command on your system, see "Command Locations" on page 25.

9 **(Optional) Enable the Common Agent Container,** cacao**, to restart when the operating system reboots.**

```
root# cacaoadm enable
```

Next Steps After installing the software, see "Environment Variables" on page 65.

▼ To Troubleshoot Directory Service Control Center Access

Use this procedure on the host where you installed Directory Service Control Center.

You must be root to perform this procedure.

1   Verify that Directory Service Control Center has been initialized properly.

```
root# /opt/SUNWdsee/dscc6/bin/dsccsetup status
***
DSCC Application is registered in Sun Java (TM) Web Console
***
DSCC Agent is registered in Cacao
***
DSCC Registry has been created
Path of DSCC registry is /var/opt/SUNWdsee/dscc6/dcc/ads
Port of DSCC registry is 3998
***
```

The default installation path for native packages on Solaris operating systems is
/opt/SUNWdsee. For the default installation path on your operating system, see "Default Paths"
on page 23.

Fix Directory Service Control Center initialization problems with the dsccsetup(1M)
command.

2   Start Java Web Console if necessary with the smcwebserver command.

```
root# /usr/sbin/smcwebserver status
Sun Java(TM) Web Console is stopped
root# /usr/sbin/smcwebserver start
Starting Sun Java(TM) Web Console Version 3.0.2 ...
The console is running.
```

3   Check the Common Agent Container if you see errors that pertain to the DSCC Agent.

The cacaoadm(1M) man page describes the error codes that the command returns. For the exact
location of this command on your system, see "Command Locations" on page 25.

If you installed Directory Server from the zip distribution, you must run the cacaoadm
command as the user who performed the installation. Otherwise, run the command as root.

After installing Directory Server, the Common Agent Container is started automatically.
However, when you reboot, you might have to start the Common Agent Container manually.

```
root# /usr/sbin/cacaoadm status
default instance is DISABLED at system startup.
Smf monitoring process:
26129
Uptime: 0 day(s), 3:16
```

For more information about the Common Agent Container, see *Sun Java Enterprise System 5 Monitoring Guide*.

▼ To Install Only Directory Server From Native Packages

This procedure covers installation of Directory Server from native packages. You must be root to perform this procedure.

Note – If you installed Directory Service Control Center, you automatically also installed Directory Server from native packages. You can use the Directory Server software that is installed alongside DSCC to create your own additional Directory Server instances on the system.

You can also install Directory Server as non-root from the zip distribution. See "To Install Only Directory Server From the Zip Distribution" on page 54 for details.

Before You Begin   Obtain the Java Enterprise System distribution for this installation, as shown in the following figure.
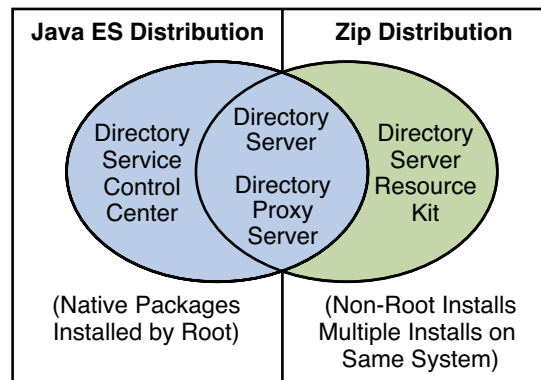
Complete the worksheet that follows for your installation.

Requisite Information	Hints	Your Answers
Fully qualified hostname of the system where you install Directory Server	Example: ds.example.com	

Requisite Information	Hints	Your Answers
(Optional) Cacao common agent container port number to access from Directory Service Control Center	Default: `11162`	
File system paths where you create Directory Server instances	Example: `/local/ds/` Create instances only on local file systems, never on network–mounted file systems such as NFS. Each path is henceforth referred to as an *instance-path.*	
LDAP port number	Default: 389 when installing as `root`; 1389 for non-`root`	
LDAP/SSL port number	Default: 636 when installing as `root`; 1636 for non-`root`	
Directory Manager DN	Default: `cn=Directory Manager`	
Directory Manager password	Must be at least 8 characters long	
Base suffix DN	Example: `dc=example,dc=com`	
(UNIX systems) Server user (`uid`)	Example: `noaccess`	
(UNIX systems) Server group (`gid`)	Example: `noaccess`	

1   Install prerequisite patches or service packs for your platform.

See "Operating System Requirements" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* for details.

2   Using the Java Enterprise System distribution, run the Java ES installer as `root`**.**

root# `./installer`

3   Select the Directory Server component for installation.

4 Choose to configure the software later, as you will register the software and create server instances after installation.

5   **Complete installation with the Java ES installer.**

At this point, you can now create server instances on the system. See "Server Instance Creation" on page 66 for details.

6   **(Optional) Enable the Common Agent Container,** cacao**, to restart when the operating system reboots.**

```
root# cacaoadm enable
```

Next Steps   After installing the software, see "Environment Variables" on page 65.

▼ To Install Only Directory Server From the Zip Distribution

This procedure covers installation of the software required to create Directory Server instances on the system.

Non-root users can install this component from the zip distribution.

On Windows systems in order to allow remote management capabilities when installing the zip distribution, you must install either as Administrator, or as a user who belongs directly to the local Administrators group.

You can also install Directory Server from native packages. See "To Install Only Directory Server From Native Packages" on page 51 for details.

Before You Begin   Obtain the zip distribution for this installation.

As shown in the following illustration, the zip distribution does not include Directory Service Control Center. You must install Directory Service Control Center from the native package Java ES distribution, as described in "To Install Directory Service Control Center From Native Packages" on page 43.

Java ES Distribution	Zip Distribution
Directory Service Control Center	Directory Server Directory Proxy Server	Directory Server Resource Kit
(Native Packages Installed by Root)	(Non-Root Installs Multiple Installs on Same System)

Complete the worksheet that follows for your installation.

Requisite Information	Hints	Your Answers
Fully qualified hostname of the system where you install Directory Server	Example: ds.example.com	
(Optional) Cacao common agent container port number to access from Directory Service Control Center	Default: 11162	
File system paths where you create Directory Server instances	Example: /local/ds/ Create instances only on local file systems, never on network–mounted file systems such as NFS. Each path is henceforth referred to as an *instance-path*.	
LDAP port number	Default: 389 when installing as root; 1389 for non-root	
LDAP/SSL port number	Default: 636 when installing as root; 1636 for non-root	
Directory Manager DN	Default: cn=Directory Manager	
Directory Manager password	Must be at least 8 characters long	
Base suffix DN	Example: dc=example,dc=com	

Requisite Information	Hints	Your Answers
(UNIX systems) Server user (uid)	Example: noaccess	
(UNIX systems) Server group (gid)	Example: noaccess	

The default user and group IDs for zip installations are those of the user performing the installation.

1 Install prerequisite patches or service packs for your platform.

See "Operating System Requirements" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* for details.

2 Change to the zip distribution directory containing the dsee_deploy **command.**

3 Install the software with the dsee_deploy**(1M) command.**

```
$ ./dsee_deploy install -c DS -i install-path options
```

For example, the following command installs the component under /local, assuming you have access to create a file system directory there.

```
$ ./dsee_deploy install -c DS -i /local
```

You can also use the --no-inter option to install in non-interactive mode, accepting the license without confirmation. Non-interactive mode is particularly useful for silent installation.

This step installs a Common Agent Container, cacao, with the local Directory Service Control Center agent as well, allowing you to use DSCC to create server instances. *The previous command works properly only if you have not yet installed a Common Agent Container using the default port*, 11162.

If you installed DSCC from native packages previously on the same system, a Common Agent Container using the default port is already installed. Specify a different port using the -p option.

```
$ ./dsee_deploy install -c DS -i /local -p 11169
```

At this point, you can now create server instances on the system. See "Server Instance Creation" on page 66 for details.

Next Steps   After installing the software, see "Environment Variables" on page 65.

▼ To Install Only Directory Proxy Server From Native Packages

This procedure covers installation of Directory Proxy Server from native packages. You must be root to perform this procedure.

You can also install Directory Proxy Server as non-root from the zip distribution. See "To Install Only Directory Proxy Server From the Zip Distribution" on page 60 for details.

Before You Begin Obtain the Java Enterprise System distribution for this installation, as shown in the following figure.

Complete the worksheet that follows for your installation.

Requisite Information	Hints	Your Answers
Fully qualified hostname of the system where you install Directory Proxy Server	Example: dps.example.com	
(Optional) Cacao common agent container port number to access from Directory Service Control Center	Default: 11162	

Requisite Information	Hints	Your Answers
File system paths where you create Directory Proxy Server instances	Example: `/local/dps/` Create instances only on local file systems, never on network–mounted file systems such as NFS. Each path is henceforth referred to as an *instance-path.*	
LDAP port number	Default: 389 when installing as `root`; 1389 for non-`root`	
LDAP/SSL port number	Default: 636 when installing as `root`; 1636 for `root`	
Directory Proxy Manager DN	Default: `cn=Proxy Manager`	
Directory Proxy Manager password	Must be at least 8 characters long	
(UNIX platforms) Server user (`uid`)	Example: `noaccess`	
(UNIX platforms) Server group (`gid`)	Example: `noaccess`	
(Optional) Connection information for each server to access through the proxy	Example: `ds1.example.com:1389,` `ds2.example.com:1636`	

1 Install prerequisite patches or service packs for your platform.

See "Operating System Requirements" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* for details.

2 Using the Java Enterprise System distribution, run the Java ES installer as `root`**.**

```
root# ./installer
```

3 Select the Directory Proxy Server component for installation.

4 Choose to configure the software later, as you will register the software and create server instances after installation.

5   **Complete installation with the Java ES installer.**

At this point, you can now create server instances on the system. See "Server Instance Creation" on page 66 for details.

6   **(Optional) Enable the Common Agent Container,** `cacao`**, to restart when the operating system reboots.**

```
root# cacaoadm enable
```

Next Steps   After installing the software, see "Environment Variables" on page 65.

▼ To Install Only Directory Proxy Server From the Zip Distribution

This procedure covers installation of the software required to create Directory Proxy Server instances on the system.

Non-root users can install this component from the zip distribution.

On Windows systems in order to allow remote management capabilities when installing the zip distribution, you must install either as Administrator, or as a user who belongs directly to the local Administrators group.

You can also install Directory Proxy Server from native packages. See "To Install Only Directory Proxy Server From Native Packages" on page 57 for details.

Before You Begin   Obtain the zip distribution for this installation.

As shown in the following illustration, the zip distribution does not include Directory Service Control Center. You must install Directory Service Control Center from the native package Java ES distribution, as described in "To Install Directory Service Control Center From Native Packages" on page 43.

Complete the worksheet that follows for your installation.

Requisite Information	Hints	Your Answers
Fully qualified hostname of the system where you install Directory Proxy Server	Example: dps.example.com	
(Optional) Cacao common agent container port number to access from Directory Service Control Center	Default: 11162	
File system paths where you create Directory Proxy Server instances	Example: /local/dps/ Create instances only on local file systems, never on network–mounted file systems such as NFS. Each path is henceforth referred to as an *instance-path*.	
LDAP port number	Default: 389 when installing as root; 1389 for non-root	
LDAP/SSL port number	Default: 636 when installing as root; 1636 for root	
Directory Proxy Manager DN	Default: cn=Proxy Manager	
Directory Proxy Manager password	Must be at least 8 characters long	
(UNIX platforms) Server user (uid)	Example: noaccess	

Requisite Information	Hints	Your Answers
(UNIX platforms) Server group (gid)	Example: noaccess	
(Optional) Connection information for each server to access through the proxy	Example: ds1.example.com:1389, ds2.example.com:1636	

The default user and group IDs for zip installations are those of the user performing the installation.

1   Install prerequisite patches or service packs for your platform.

See "Operating System Requirements" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* for details.

2   Change to the zip distribution directory containing the dsee_deploy **command.**

3   Install the software with the dsee_deploy**(1M) command.**

$ **./dsee_deploy install -c DPS -i** *install-path options*

For example, the following command installs the component under /local, assuming you have access to create a file system directory there.

$ **./dsee_deploy install -c DPS -i /local**

You can also use the --no-inter option to install in non-interactive mode, accepting the license without confirmation. Non-interactive mode is particularly useful for silent installation.

This step installs a Common Agent Container, cacao, with the local Directory Service Control Center agent as well, allowing you to use DSCC to create server instances. *The previous command works properly only if you have not yet installed a Common Agent Container using the default port*, 11162.

If you installed DSCC from native packages previously on the same system, a Common Agent Container using the default port is already installed. Specify a different port using the -p option.

$ **./dsee_deploy install -c DPS -i /local -p 11169**

At this point, you can now create server instances on the system. See "Server Instance Creation" on page 66 for details.

Next Steps   After installing the software, see "Environment Variables" on page 65.

▼ To Install Only Directory Server Resource Kit From the Zip Distribution

This procedure covers installation of the software required to use Directory Server Resource Kit tools.

Non-root users can install this component from the zip distribution.

Before You Begin   Obtain the zip distribution for this installation.
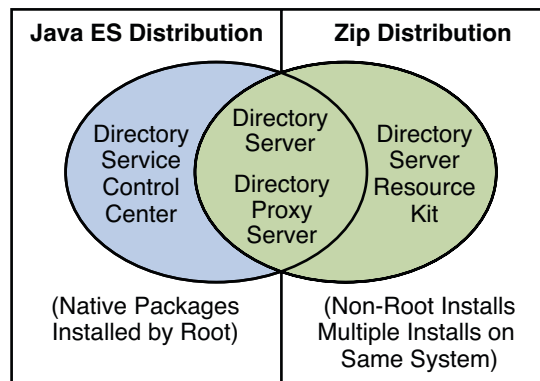
Complete the worksheet that follows for your installation.

Requisite Information	Hints	Your answers
File system path where you install developer tools	This path is henceforth referred to as *install-path*.	
(Optional) URL to Application Server or Web Server administration interface used to deploy the NameFinder application	Examples: http://localhost:4848, http://localhost:8888	
(Optional) Hostname and port number for the LDAP directory or directories queried by the NameFinder application	Example: ds.example.com:389	
(Optional) Base DN of suffix queried by the NameFinder application	Example: dc=example,dc=com	

1   Install prerequisite patches or service packs for your platform.

See "Operating System Requirements" in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes* for details.

2   Change to the zip distribution directory containing the dsee_deploy **command.**

3   Install the software with the dsee_deploy**(1M) command.**

```
$ ./dsee_deploy install -c DSRK -i install-path options
```

For example, the following command installs the component under /local, assuming you have access to create a file system directory there.

```
$ ./dsee_deploy install -c DSRK -i /local
```

You can also use the --no-inter option to install in non-interactive mode, accepting the license without confirmation. Non-interactive mode is particularly useful for silent installation.

4   (Optional) Load sample data in your directory.

Examples that use the command line tools depend on sample data residing under the dc=example,dc=com suffix of your directory.

You can set up part of the data that are required by creating a dc=example,dc=com suffix. You can then populating the suffix with entries from the ldif/Example.ldif file.

Read the Example.ldif file to find bind passwords needed in the examples.

After you load the Example.ldif content into the directory, generate test data for examples with the makeldif(1) command and the following template:

```
define suffix=dc=example,dc=com
define maildomain=example.com

branch: ou=test,[suffix]
subordinateTemplate: person:100

template: person
rdnAttr: uid
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
givenName: <first>
sn: <last>
cn: {givenName} {sn}
initials: {givenName:1}{sn:1}
employeeNumber: <sequential>
uid: test{employeeNumber}
mail: {uid}@[maildomain]
userPassword: auth{employeeNumber}{employeeNumber}
```

```
telephoneNumber: <random>
description: This is the description for {cn}.
```

First, copy the template content to template.ldif. Next, use commands such as the following to generate the data in test.ldif, and to load the content into the directory.

The following commands generate a test.ldif file and add its content to the directory, as long as the directory already contains the sample entries from Example.ldif.

```
$ cd install-path/dsrk6/bin/example_files/
$ makeldif -t test.template -o test.ldif
Processing complete.
101 total entries written.
$ ldapmodify -a -D uid=hmiller,dc=example,dc=com -w - -f test.ldif
Enter bind password:
...
```

If you read Example.ldif, you see that the password for hmiller is hillock.

Next Steps After installing the software, see "Environment Variables" on page 65.

Environment Variables

This section lists environment variables that you can set to facilitate creating server instances and using Directory Server Resource Kit and software development kits.

Environment Variable	Set to include…	Applies to…
DIR_PROXY_HOST	Hostname of Directory Proxy Server for administration tools	dpconf(1M) command
DIR_PROXY_PORT	Port number of Directory Proxy Server for administration tools	dpconf(1M) command
DIRSERV_HOST	Hostname of Directory Server for administration tools	dsconf(1M) command
DIRSERV_PORT	Port number of Directory Server or for administration tools	dsconf(1M) command
LDAP_ADMIN_PWF	Path to the file that contains the directory administrator password To administer all servers registered with Directory Service Control Center, set this environment variable to a file containing Directory Service Manager password.	dpconf(1M), dsconf(1M) commands

Environment Variable	Set to include…	Applies to…
LDAP_ADMIN_USER	Directory administrator DN To administer all servers registered with Directory Service Control Center, set this environment variable to `cn=admin,cn=Administrators,cn=dscc`.	`dpconf(1M)`, `dsconf(1M)` commands
MANPATH	*install-path*/dsee6/man	Online manual pages to browse with the man command
MANSECT	Add any of the following sections not in your MANSECT environment variable. `1:1m:4:5dsconf:5dpconf:5dssd:5dsat:5dsoc` Alternatively, specify the sections to search explicitly when using the man command.	The man command can use the MANSECT environment variable to identify the sections to search by default.
PATH	*install-path*/dps6/bin	Directory Proxy Server commands
	install-path/ds6/bin	Directory Server commands
	install-path/dscc6/bin	Directory Service Control Center commands
	install-path/dsrk6/bin	Directory Server Resource Kit and LDAP client commands

Server Instance Creation

After installing server software as described in "Software Installation" on page 42, create server instances according to the procedures in this section.

- "To Create a Directory Server Instance With DSCC" on page 66
- "To Create a Directory Server Instance From the Command Line" on page 67
- "To Create a Directory Proxy Server Instance With DSCC" on page 71
- "To Create a Directory Proxy Server Instance From the Command Line" on page 72

▼ To Create a Directory Server Instance With DSCC

Non-root users can create server instances.

Before You Begin   Install the component software as described in "Software Installation" on page 42.

1   **Access Directory Service Control Center through Java Web Console.**
The default URL for Java Web Console on the local system is `https://localhost:6789`.

2 Follow the instructions in the Directory Service Control Center New Server wizard to create the server instance.

▼ To Create a Directory Server Instance From the Command Line

In this procedure, you create a *server instance* on the local host using the dsadm command. You then create a suffix that you populate with data using the dsconf command.

Non-root users can create server instances.

A Directory Server instance contains the configuration and data necessary to respond to directory client applications. When you start or stop an instance, you start or stop the server process. The server process is what serves directory client requests corresponding to the data managed by that instance.

The dsadm command enables you to manage a Directory Server instance and the files belonging to that instance on the local host. The command does not let you administer servers over the network, but only directly on the local host. The dsadm command has subcommands for each key management task. For a complete description, see dsadm(1M).

The dsconf command is an LDAP client. The command enables you to configure nearly all server settings on a running Directory Server instance from the command line. You can configure settings whether the server is on the local host or another host that is accessible across the network. The dsconf command has subcommands for each key configuration task. For a complete description, see dsconf(1M).

Before You Begin Install the component software, then set your PATH as described in "Software Installation" on page 42.

1 Create a new Directory Server instance.

$ **dsadm create -p** *port* **-P** *SSL-port instance-path*

For example, the following command creates an instance under the existing directory, /local/, in a new directory, /local/ds/. The new instance has default ports 389 for LDAP, 636 for LDAPS for root, and 1389 for LDAP, 1636 for LDAPS for non-root users.

```
$ dsadm create /local/ds
Choose the Directory Manager password:
Confirm the Directory Manager password:
Use 'dsadm start /local/ds' to start the instance
```

Notice that the instance is created in a directory on the local file system, not a network file system.

2 Start the instance.

$ **dsadm start** *instance-path*

For example, the following command starts the instance located under /local/ds/.

```
$ dsadm start /local/ds
Server started: pid=2845
```

3 Verify that you can read the root DSE of the new instance.

```
$ ldapsearch -h localhost -p 1389 -b "" -s base "(objectclass=*)"
version: 1
dn:
objectClass: top
...
supportedLDAPVersion: 3
vendorName: Sun Microsystems, Inc.
vendorVersion: Sun-Java(tm)-System-Directory/6.0
...
```

Note – At this point in the procedure, you have a working server instance. You must further configure the server instance, however. Furthermore, the instance is not yet registered with Directory Service Control Center.

---

4 (Optional) Use the new password policy mode, unless the instance belongs to a topology with version 5 instances.

Your server instance might be standalone. Alternatively, your instance might belong to a replication topology that has already been migrated to the new password policy mode. In either case, perform this step.

```
$ dsconf pwd-compat -h localhost -p 1389 to-DS6-migration-mode
Certificate "CN=hostname, CN=1636, CN=Directory Server, O=Sun Microsystems"
 presented by the server is not trusted.
Type "Y" to accept, "y" to accept just once, "n" to refuse, "d" for more details: Y
Enter "cn=Directory Manager" password:
## Beginning password policy compatibility changes.
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
$ dsconf pwd-compat -p 1389 to-DS6-mode
Enter "cn=Directory Manager" password:
## Beginning password policy compatibility changes.
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
```

5 (Optional) Prepare an example suffix.

a. Create an empty suffix.

For example, the following command creates a suffix with root dc=example,dc=com.

```
$ dsconf create-suffix -h localhost -p 1389 dc=example,dc=com
Enter "cn=Directory Manager" password:
$
```

b. Populate the suffix with LDIF data.

If you plan to populate the suffix with data that is replicated from another Directory Server instance, skip this step.

For example, the following command fills the suffix that you created with sample data from Example.ldif.

```
$ dsconf import -h localhost -p 1389 install-path/ds6/ldif/Example.ldif dc=example,dc=com
Enter "cn=Directory Manager" password:
New data will override existing data of the suffix "dc=example,dc=com".
Initialization will have to be performed on replicated suffixes.
Do you want to continue [y/n] ?  y
```

Chapter 1 • Installation

```
## Index buffering enabled with bucket size 40
## Beginning import job...
## Processing file "install-path/ds6/ldif/Example.ldif"
## Finished scanning file "install-path/ds6/ldif/Example.ldif" (160 entries)
## Workers finished; cleaning up...
## Workers cleaned up.
## Cleaning up producer thread...
## Indexing complete.
## Starting numsubordinates attribute generation.
 This may take a while, please wait for further activity reports.
## Numsubordinates attribute generation complete. Flushing caches...
## Closing files...
## Import complete.  Processed 160 entries in 4 seconds. (40.00 entries/sec)

Task completed (slapd exit code: 0).
```

c. Verify that you can find an entry in the new instance.

```
$ ldapsearch -h localhost -p 1389 -b dc=example,dc=com "(uid=bjensen)"
version: 1
dn: uid=bjensen, ou=People, dc=example,dc=com
cn: Barbara Jensen
cn: Babs Jensen
sn: Jensen
givenName: Barbara
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Product Development
ou: People
l: Cupertino
uid: bjensen
mail: bjensen@example.com
telephoneNumber: +1 408 555 1862
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 0209
```

6 (Optional) Register the server instance with Directory Service Control Center by using either of the following methods.

- **Login to DSCC, and then use the Register Existing Server action on the Servers tab of the Directory Servers tab.**

  Access DSCC through the URL https://localhost:6789.

- **Use the command** dsccreg add-server.

  ```
  $ dsccreg add-server -h dscchost --description "My DS" /local/ds
  Enter DSCC administrator's password:
  ```

```
/local/ds is an instance of DS
Enter password of "cn=Directory Manager" for /local/ds:
This operation will restart /local/ds.
Do you want to continue ? (y/n) y
Connecting to /local/ds
Enabling DSCC access to /local/ds
Restarting /local/ds
Registering /local/ds in DSCC on dscchost.
```

See dsccreg(1M) for more information about the command.

7   **(Optional) If you installed from native packages with the Java Enterprise System distribution, enable the server to restart when the operating system reboots.**

On Solaris 10 and Windows systems, use the dsadm enable-service command.

```
root# dsadm enable-service /local/ds
```

On Solaris 9, HP-UX, and Red Hat systems, use the dsadm autostart command.

```
root# dsadm autostart /local/ds
```

If you installed with the zip distribution, this step must be done manually, with a script run at system startup time, for example.

Next Steps   At this point, you can add more suffixes, configure replication with other server instances, tune the instance, and generally proceed with other configuration operations.

See the online help inside Directory Service Control Center for hints on configuring Directory Server through the graphical user interface.

See Part I, "Directory Server Administration," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide* for instructions on configuring Directory Server with command-line administration tools.

▼ **To Create a Directory Proxy Server Instance With DSCC**

Non-root users can create server instances.

Before You Begin   Install the component software as described in "Software Installation" on page 42.

1   **Access Directory Service Control Center through Java Web Console.**
The default URL for Java Web Console on the local system is https://localhost:6789.

2   **Follow the instructions in the Directory Service Control Center New Server wizard to create the server instance.**

▼ To Create a Directory Proxy Server Instance From the Command Line

In this procedure, you create a *server instance* on the local host using the dpadm command. You then configure the instance using the dpconf command.

Non-root users can create server instances.

A Directory Proxy Server instance must be configured to proxy directory client application requests to *data sources* through *data views*. When you start or stop an instance, you start or stop the server process that proxies directory client application requests.

The dpadm command enables you to manage a Directory Proxy Server instance and the files belonging to that instance on the local host. The command does not allow you to administer

servers over the network, but only directly on the local host. The dpadm command has subcommands for each key management task. For a complete description, see dpadm(1M).

The dpconf command is an LDAP client. The command enables you to configure nearly all server settings on a running Directory Proxy Server instance from the command line. You can configure settings whether the server is on the local host or another host that is accessible across the network. The dpconf command has subcommands for each key configuration task. For a complete description, see dpconf(1M).

Before You Begin Install the component software, then set your PATH as described in "Software Installation" on page 42.

1 Create a new Directory Proxy Server instance.

```
$ dpadm create -p port -P SSL-port instance-path
```

For example, the following command creates an instance under the existing directory, /local/, in a new directory, /local/dps/. The default ports are 389 for LDAP, 636 for LDAPS for root, and 1389 for LDAP, 1636 for LDAPS for non-root users.

```
$ dpadm create -p 1390 -P 1637 /local/dps
Choose the Proxy Manager password:
Confirm the Proxy Manager password:
Use 'dpadm start /local/dps' to start the instance
```

Notice that the instance must be created in a directory on the local file system, not a network file system.

2 Start the instance.

```
$ dpadm start instance-path
```

For example, the following command starts the instance located under /local/dps/.

```
$ dpadm start /local/dps
...
Directory Proxy Server instance '/local/dps' started: pid=28732
```

3 Verify that you can read the root DSE of the new instance.

```
$ ldapsearch -h localhost -p 1390 -b "" -s base "(objectclass=*)"
version: 1
dn:
objectClass: top
objectClass: extensibleObject
supportedLDAPVersion: 2
supportedLDAPVersion: 3
...
vendorName: Sun Microsystems, Inc
```

```
vendorVersion: Directory Proxy Server 6.0
...
```

Note – At this point in the procedure, you have a working server instance. You must further configure the server instance, however. Furthermore, the instance is not yet registered with Directory Service Control Center.

---

4 (Optional) Enable the Directory Proxy Server instance to function as an LDAP proxy.

a. Create an LDAP data source.

For example, the following command creates a data source, My DS, pointing to the directory instance created on the local host in .

```
$ dpconf create-ldap-data-source -h localhost -p 1390 "My DS" localhost:1389
Certificate "CN=hostname:1390" presented by the server is not trusted.
Type "Y" to accept, "y" to accept just once, "n" to refuse, "d" for more details: Y
Enter "cn=Proxy Manager" password:
```

b. Create an LDAP data source pool.

```
$ dpconf create-ldap-data-source-pool -h localhost -p 1390 "My Pool"
Enter "cn=Proxy Manager" password:
```

c. Attach the LDAP data source to the LDAP data source pool.

```
$ dpconf attach-ldap-data-source -h localhost -p 1390 "My Pool" "My DS"
Enter "cn=Proxy Manager" password:
```

d. Create an LDAP data view into the LDAP data source pool.

For example, the following command creates a data view, My View, which allows client applications to view the suffix dc=example,dc=com:

```
$ dpconf create-ldap-data-view -h localhost -p 1390 "My View" \
 "My Pool" dc=example,dc=com
Enter "cn=Proxy Manager" password:
```

e. Enable the LDAP data source, then restart the server for the change to take effect.

```
$ dpconf set-ldap-data-source-prop -h localhost -p 1390 "My DS" is-enabled:true
Enter "cn=Proxy Manager" password:
$ dpadm restart /local/dps
Directory Proxy Server instance '/local/dps' stopped
[31/Aug/2006:11:32:26 +0200] - STARTUP   - INFO  -
 Sun Java(TM) System Directory Proxy Server/6.0 (Build 0824060144) starting up
Directory Proxy Server instance '/local/dps' started: pid=28901
```

f. Enable searches on the LDAP data source.

```
$ dpconf set-attached-ldap-data-source-prop -h localhost -p 1390 \
 "My Pool" "My DS" search-weight:100
Enter "cn=Proxy Manager" password:
```

g. Verify that you can read directory data through the new instance.

```
$ ldapsearch -h localhost -p 1390 -b dc=example,dc=com "(uid=bjensen)"
version: 1
dn: uid=bjensen, ou=People, dc=example,dc=com
cn: Barbara Jensen
cn: Babs Jensen
sn: Jensen
givenName: Barbara
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Product Development
ou: People
l: Cupertino
uid: bjensen
mail: bjensen@example.com
telephoneNumber: +1 408 555 1862
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 0209
```

Note – Notice that LDAP search operations work for the suffix handled by your data view, but do not work for other suffixes. If you search a suffix for which no data view is configured, the server returns an error.

```
$ ldapsearch -h localhost -p 1390 -b o=example.com "(uid=bjensen)"
ldap_search: Operations error
ldap_search: additional info: Unable to retrieve a backend SEARCH
 connection to process the search request
```

For instructions detailed instructions on configuring Directory Proxy Server, see Part II, "Directory Proxy Server Administration," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

5 **(Optional) Register the server instance with Directory Service Control Center by using either of the following methods.**

- **Login to DSCC, and then use the Register Existing Server action on the Proxy Servers tab.**
  Access DSCC through the URL https://localhost:6789.

■ **Use the command** dsccreg add-server**.**

```
$ dsccreg add-server -h dscchost --description "My Proxy" /local/dps
Enter DSCC administrator's password:
/local/dps is an instance of DPS
Enter password of "cn=Proxy Manager" for /local/dps:
Connecting to /local/dps
Enabling DSCC access to /local/dps
Registering /local/dps in DSCC on dscchost.
```

See dsccreg(1M) for more information about the command.

6 **(Optional) If you installed from native packages with the Java Enterprise System distribution, enable the server to restart when the operating system reboots.**

On Solaris 10 and Windows systems, use the dpadm enable-service command.

```
root# dpadm enable-service /local/dps
```

On Solaris 9, HP-UX, and Red Hat systems, use the dpadm autostart command.

```
root# dpadm autostart /local/dps
```

If you installed with the zip distribution, this step must be done manually, with a script run at system startup time, for example.

Next Steps At this point, you can continue to configure further data sources and data views. You can also configure load balancing, data distribution, and other server capabilities.

See the online help inside Directory Service Control Center for hints on configuring Directory Proxy Server through the graphical user interface.

See Part II, "Directory Proxy Server Administration," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide* for instructions on configuring Directory Proxy Server with command-line administration tools.

Working With Sun Cryptographic Framework on Solaris 10 Systems

This section explains briefly how to use Sun Crypto Accelerator cards through the Sun cryptographic framework on Solaris 10 systems with Directory Server, and Directory Proxy Server. See Chapter 13, "Solaris Cryptographic Framework (Overview)," in *System Administration Guide: Security Services* for more information about the framework.

▼ To Use Directory Server With Cryptographic Hardware on a Solaris 10 System

Before You Begin   This procedure is designed for use with Sun Crypto Accelerator hardware. Perform the following procedure as the same user who runs the Directory Server instance.

1 **Set the PIN used to access the cryptographic framework with the** `pktool setpin` **command.**

2 **Export the current Directory Server certificate to a PKCS#12 file.**

The following command shows how to perform this step if the Directory Server instance is located under `/local/ds/`.

```
$ dsadm export-cert -o cert-file /local/ds defaultCert
```

3 **Configure Directory Server to use the appropriate token when accessing the key material.**

Typically, the token is `Sun Metaslot`.

```
$ dsconf set-server-prop 'ssl-rsa-security-device:Sun Metaslot'
```

4 **Stop Directory Server.**

```
$ dsadm stop /local/ds
```

5 **(Optional) If you have no other certificates in the existing certificate database for the Directory Server instance, remove the certificate database.**

```
$ rm -f /local/ds/alias/*.db
```

This optional step ensures that no certificates are stored in the software database.

6 **Create a new certificate database backed by the Solaris cryptographic framework.**

If you did not remove the certificate database, you do not need to run the `modutil -create` line in this example.

```
$ /usr/sfw/bin/64/modutil -create -dbdir /local/ds/alias -dbprefix slapd-
$ /usr/sfw/bin/64/modutil -add "Solaris Kernel Crypto Driver" -libfile \
 /usr/lib/64/libpkcs11.so -dbdir /local/ds/alias -dbprefix slapd-
$ /usr/sfw/bin/64/modutil -enable "Solaris Kernel Crypto Driver" \
 -dbdir /local/ds/alias -dbprefix slapd-
```

7 **Import the PKCS#12 certificate that you exported.**

```
$ /usr/sfw/bin/64/pk12util -i cert-file \
 -d /local/ds/alias -P slapd- -h "Sun Metaslot"
$ /usr/sfw/bin/64/certutil -M -n "Sun Metaslot:defaultCert" -t CTu \
 -d /local/ds/alias -P slapd-
```

If your accelerator board has a FIPS 140-2 keystore, make sure the private key is generated on the device. Sun Crypto Accelerator 4000 and 6000 boards have FIPS 140-2 keystores, for example. The exact process depends on the board.

8   Create a password file that contains the PIN needed to access the cryptographic framework.

```
$ echo "Sun Metaslot:password" > /local/ds/alias/slapd-pin.txt
```

9   Start Directory Server.

```
$ dsadm start /local/ds
```

▼ To Use Directory Proxy Server With Cryptographic Hardware on a Solaris 10 System

Before You Begin   This procedure is designed for use with Sun Crypto Accelerator hardware. Perform the following procedure as the same user who runs the Directory Proxy Server instance.

1   Stop Directory Proxy Server.

```
$ dpadm stop /local/dps
```

2   Turn off certificate database password storage.

```
$ dpadm set-flags /local/dps cert-pwd-prompt=on
Choose the certificate database password:
Confirm the certificate database password:
```

3   Set the PIN used to access the cryptographic framework with the pktool setpin **command.**

Use the same password that you entered when turning off certificate database password storage.

4   Generate a key pair, using the cryptographic framework as the key store.

```
$ keytool -genkeypair -alias defaultDPScert
 -dname "ou=dps server,dc=example,dc=com" -keyalg RSA -sigalg MD5withRSA
 -validity 3652 -storetype PKCS11 -keystore NONE -storepass pin-password
```

Here, *pin-password* is the password you set as the PIN with the pktool setpin command.

5   Edit the Directory Proxy Server configuration file, adding the following attributes to the base entry, cn=config**.**

```
serverCertificateNickName: defaultDPScert
certificateKeyStore: NONE
certificateKeyStoreType: PKCS11
```

6   Start Directory Proxy Server.

```
$ dpadm start /local/dps
```

2

Uninstallation

This chapter guides you in removing Directory Server Enterprise Edition software.

- "Server Instance Removal" on page 79 covers removing the server instances that depend on the software to remove.
- "Software Removal" on page 81 covers how to remove the software after you have removed server instances.

Server Instance Removal

Before removing Directory Server Enterprise Edition software used by server instances on the system, you must remove all the server instances.

- "To Delete a Directory Proxy Server Instance With DSCC" on page 79
- "To Delete a Directory Proxy Server Instance From the Command Line" on page 80
- "To Delete a Directory Server Instance With DSCC" on page 80
- "To Delete a Directory Server Instance From the Command Line" on page 80

▼ To Delete a Directory Proxy Server Instance With DSCC

1 **Access Directory Service Control Center through Java Web Console.**

The default URL for Java Web Console on the local system is `https://localhost:6789`.

2 **Delete the server instance with the Delete command in the action drop-down list.**

▼ To Delete a Directory Proxy Server Instance From the Command Line

1 **(Optional) If you have used DSCC to manage the server instance, remove registration for the server.**

```
$ dsccreg remove-server -h dscchost /local/dps
Enter DSCC administrator's password:
/local/dps is an instance of DPS
Enter password of "cn=Proxy Manager" for /local/dps:
Unregistering /local/dps from DSCC on dscchost.
Connecting to /local/dps
Disabling DSCC access to /local/dps
```

For details, see dsccreg(1M)

2 **Delete the server instance.**

```
$ dpadm delete /local/dps
Directory Proxy Server instance '/local/dps' stopped
Directory Proxy Server instance '/local/dps' removed.
```

See Also After you have removed all server instances on the system, you can proceed to "Software Removal" on page 81.

▼ To Delete a Directory Server Instance With DSCC

Deleting a Directory Server instance *completely removes all instance files, including all directory databases managed by the instance.* Before you delete an instance, back up your data as described in Chapter 8, "Directory Server Backup and Restore," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

1 **Access Directory Service Control Center through Java Web Console.**

The default URL for Java Web Console on the local system is https://localhost:6789.

2 **Delete the server instance with the Delete command in the action drop-down list.**

▼ To Delete a Directory Server Instance From the Command Line

Deleting a Directory Server instance *completely removes all instance files, including all directory databases managed by the instance.* Before you delete an instance, back up your data as described in Chapter 8, "Directory Server Backup and Restore," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

1 **(Optional) If you have used DSCC to manage the server instance, remove registration for the server.**

```
$ dsccreg remove-server -h dscchost /local/ds
Enter DSCC administrator's password:
/local/ds is an instance of DS
Enter password of "cn=Directory Manager" for /local/ds:
This operation will restart /local/ds.
Do you want to continue ? (y/n) y
Unregistering /local/ds from DSCC on dscchost.
Connecting to /local/ds
Disabling DSCC access to /local/ds
Restarting /local/ds
```

For details, see dsccreg(1M)

2 **Delete the server instance.**

```
$ dsadm delete /local/ds
Server stopped
/local/ds deleted
```

See Also After you have removed all server instances on the system, you can proceed to "Software Removal" on page 81.

Software Removal

After you have removed all server instances that depend on the installed product components, you can remove the component software.

- "To Remove Directory Service Control Center Software" on page 81
- "To Remove Directory Server, or Directory Proxy Server Installed From Native Packages" on page 82
- "To Remove Software Installed From the Zip Distribution" on page 82
- "To Force Removal of Software Installed From the Zip Distribution" on page 83

▼ To Remove Directory Service Control Center Software

By removing all of DSCC, you also remove Directory Server packages from the system.

1 **Dismantle DSCC with the** dsccsetup dismantle **command.**

For example, on a Solaris system the following command dismantles DSCC.

```
root# /opt/SUNWdsee/dscc6/bin/dsccsetup dismantle
***
Unregistering DSCC Application from Sun Java(TM) Web Console...
```

```
This operation is going to stop Sun Java(TM) Web Console.
Do you want to continue ? [y,n] y
Stopping Sun Java(TM) Web Console...
Unregistration is on-going. Please wait...
/var/opt/SUNWdsee/dscc6/dcc has not been removed
DSCC Application has been unregistered from Sun Java(TM) Web Console
Restarting Sun Java(TM) Web Console
Please wait : this may take several seconds...
Sun Java(TM) Web Console restarted successfully
***
```

The dsccsetup command is located in *install-path*/dscc6/bin/dsccsetup. See "Default Paths" on page 23 to determine the default *install-path* for your system.

2   Remove Directory Service Control Center with the Java ES installer.

For instructions, see the Java Enterprise System documentation at http://docs.sun.com/coll/1286.2.

▼ To Remove Directory Server, or Directory Proxy Server Installed From Native Packages

◗ **Remove the software with the Java ES installer.**

For instructions, see the Java Enterprise System documentation at http://docs.sun.com/coll/1286.2.

▼ To Remove Software Installed From the Zip Distribution

◗ **Remove the software with the** dsee_deploy**(1M) command.**

If zip distribution software was installed by a non-root user, that user can also remove the software.

For example, to remove all Directory Server Enterprise Edition software installed under /local, issue the following command.

```
$ /local/dsee6/bin/dsee_deploy uninstall -c ALL -i /local
```

See Also   For a full list of supported components, see dsee_deploy(1M).

▼ To Force Removal of Software Installed From the Zip Distribution

Before You Begin   You can force removal by deleting installed files, *if you have installed the software from the zip distribution*.

If zip distribution software was installed by a non-root user, that user can also remove the software.

Do not directly delete files that are installed from native packages.

◗ **Remove components with a system command.**

```
$ rm -rf install-path
```

PART II

Installing Identity Synchronization for Windows

Sun Java System Identity Synchronization for Windows 6.0 (formerly Sun™ ONE Identity Synchronization for Windows) allows passwords and other, specified user attributes to flow between Sun Java System Directory Server and other systems.

This part of this guide explains how to install and configure Sun Java System Identity Synchronization for Windows for use in a production environment.

For the latest information about new features and about enhancements in this release of Identity Synchronization for Windows, see the *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes.*

Note – User interfaces that are depicted in this document are subject to change in future versions of the product.

This part includes the following chapters:

- Chapter 3 explains some basic concepts related to Identity Synchronization for Windows. The chapter covers product features, system components, command line utilities, system component distribution, and deployment examples.

- Chapter 4 describes the installation and configuration processes. The chapter also provides helpful information to prepare to install the product.

- Chapter 5 explains how to use the Identity Synchronization for Windows installation program. The chapter also explains how to install the Identity Synchronization for Windows Core component.

- Chapter 6 explains how to add and configure Core resources with the Console.

- Chapter 7 provides instructions for installing the Identity Synchronization for Windows Connectors and Directory Server Plug-ins.

- Chapter 8 explains how to link and resynchronize existing users for new Identity Synchronization for Windows installations.

- Chapter 9 explains how to remove Identity Synchronization for Windows, including how to prepare for the uninstallation and how to uninstall the Console manually.

- Chapter 10 describes how to configure a secure system. This chapter covers how to harden security, secure replicated configurations, enable SSL, and add Active Directory CA certificates to certificate databases.

- Chapter 11 provides information about audit and error logging. This chapter includes instructions on how to set logging levels, how to view and understanding your log files, and directory source status.

- Chapter 12 provides information that you can use to troubleshoot your Identity Synchronization for Windows installation.

- Appendix A lists command line utilities to perform different tasks.

- Appendix B provides a sample Linkusers XML document, `linkusers-simple.cfg`. You can use the example to customize to your environment.

- Appendix C: Explains how to run Identity Synchronization for Windows services as a `non-root` user.

- Appendix D: Provides information about Synchronization User List definitions and multiple domain configurations.

- Appendix E provides an overview of how to configure a multimaster replication (MMR) deployment.

For help to install Directory Proxy Server, Directory Server, and Directory Server Resource Kit, see Part I

3

Understanding the Product

Identity Synchronization for Windows provides bidirectional password and user attributes synchronization between the Sun Java System Directory Server 6.0 and the following:

- Windows 2000 or Windows 2003 Server Active Directory
- Windows NT SAM Registry

Identity Synchronization for Windows handles synchronization events:

- **Securely**: Identity Synchronization for Windows never sends passwords "in the clear," and restricts system access to administrators only.

- **Robustly**: Identity Synchronization for Windows keeps directories synchronized— even when individual components are temporarily unavailable.

- **Efficiently**: Identity Synchronization for Windows synchronization methods place very little load on your directory servers.

Before you install (or migrate to) Sun Java System Identity Synchronization for Windows version 6.0, you should become familiar with the concepts described in this chapter, which consists of the following sections:

Product Features

Identity Synchronization for Windows provides the following features and functionality:

- **Bidirectional password synchronization**: Enables you to synchronize user passwords between the following directory sources:
  - Sun Java System Directory Server and Windows Active Directory
  - Sun Java System Directory Server and Windows NT

  Synchronizing passwords allows users to access applications using these directory sources for login authentication so they only have to remember a single password. In addition, when users have to apply periodic password updates, they only have to update their password in one environment.

- **Bidirectional user attributes synchronization** : Enables you to create, modify, and delete selected attributes in one directory environment and propagate the values automatically to the other directory environment.

- **Bidirectional user account creation synchronization** : Enables you to create or delete a user account in one directory environment and automatically propagate the new account to the other directory environment.

- **Bidirectional group synchronization**: Enables you to create or delete a group, and associate or disassociate users with that group in a directory environment. The changes you make in one directory environment automatically propagate to the other directory environment.

- **Bidirectional object deletions, activations, and inactivations**: Enables you to control the flow of object deletions and object activations and inactivations between Directory Server and Active Directory sources.

- **Bidirectional account lockout and unlockout synchronization**: Enables you to synchronize the account lockout and unlockout between the Directory Server and Active Directory sources.

- **Synchronization with multiple domains**: Enables you to synchronize with multiple Active Directory and Windows NT domains, and with multiple Active Directory forests.

- **Centralized system auditing**: Enables you to monitor installation and configuration status, the day-to-day system operations, and any error conditions related to your deployment from a single, centralized location.

You will not be required to modify entries in Windows directories, or to change the applications using the directories.

If you are using Identity Synchronization for Windows to synchronize between Directory Server and Active Directory, you need not install any components in the Windows operating environment.

If you are synchronizing between Directory Server and Windows NT, you must install the product's NT component in the Windows NT environment.

Note – The following features are not available for Windows NT:

- Bidirectional group synchronization
- Bidirectional object deletions, activations, and inactivations
- Bidirectional account lockout and unlockout synchronization

System Components

Identity Synchronization for Windows consists of a set of Core components and any number of individual *connectors* and *connector subcomponents* that allow for the synchronization of password and user attribute updates between Sun Java System Directory Server and Windows directories.
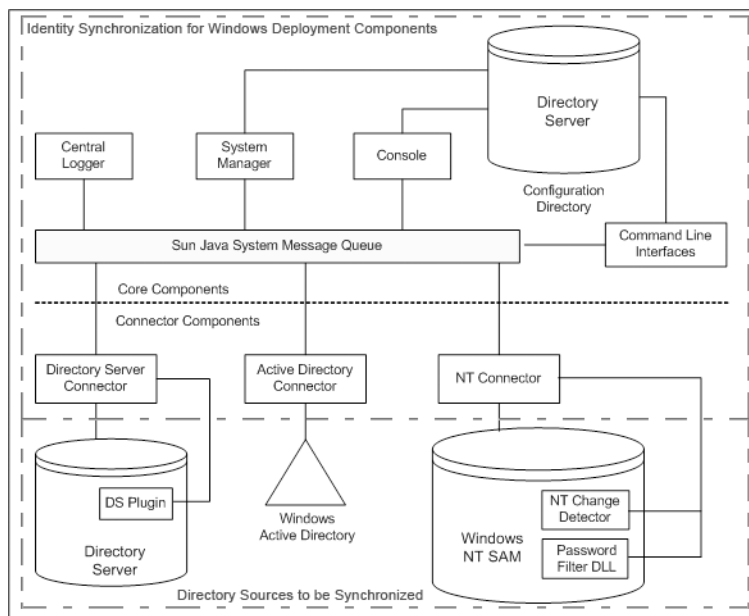
FIGURE 3–1   System Components

This section defines and describes each of the Identity Synchronization for Windows components and is organized as follows:

- "Watchdog Process" on page 90
- "Core" on page 90
- "Connectors" on page 92
- "Connector Subcomponents" on page 93

Watchdog Process

The Watchdog is an Identity Synchronization for Windows Java process that is responsible for starting, restarting, and stopping individual background Java processes. The Watchdog launches and monitors the central logger, system manager, and connectors (but does not monitor subcomponents, Message Queue, or the Identity Synchronization for Windows Console).

The Watchdog is installed anywhere you install Core and it can be started as a Solaris daemon, Linux daemon, or a Windows service. (For information about starting and stopping services, see "Starting and Stopping Services" on page 219.)

Core

When you install Identity Synchronization for Windows, you install the *Core* component first, and then you configure it to match your environment.

The core component consists of the following components, which are each separate Java processes. A description each component, begins on the referenced page:

- "Configuration Directory" on page 90
- "Console" on page 91
- "Command Line Utilities" on page 91
- "System Manager" on page 91
- "Central Logger" on page 92

Configuration Directory

Identity Synchronization for Windows *stores* its configuration data in a Directory Server configuration directory (the program does not install a configuration directory).

The console, system manager, command line utility, and the installer all read and write the product's configuration data to and from the configuration directory, including:

- Installation information about each component's health
- Configuration information for every directory, domain, connector, and Directory Server Plug-in
- Connector status
- Synchronization settings that describe the direction of user or group's creations, deletions, and attribute modifications
- Attributes to be synchronized and attribute mappings between the two directory environments Active Directory and Directory Server or Windows NT and Directory Server
- Synchronization User Lists in each directory topology
- Log settings

Console

The Identity Synchronization for Windows provides a Console that centralizes all of the product's component configuration and administration tasks.

You can use the console to do the following:

- Configure directory sources to be synchronized
- Define mappings for user entry attributes to be synchronized, in addition to passwords
- Specify which users and attributes within a directory or domain topology will be (or will not be) synchronized
- Monitor system status
- Start and stop synchronization

Command Line Utilities

Identity Synchronization for Windows also provides command line utilities that enable you to perform the following tasks directly from the command line:

- Display certificate information based on your configuration and SSL settings
- Change the Identity Synchronization for Windows configuration password
- Configure the Directory Server plug-in for a specified Directory Server source.
- Prepare a Sun Java System Directory Server source for use by Identity Synchronization for Windows
- Display the steps you must perform to complete the installation or configuration process and view the status of installed connectors, the system manager, and Message Queue
- Reset connector states in the configuration directory to *uninstalled*
- Synchronize and link existing users in two directories, and pre-populate directories as part of the installation process
- Enable or disable account lockout
- Enable or disable group synchronization
- Start and stop synchronization

For a detailed description of the product's command line utilities and how to use them, see Appendix A

System Manager

The Identity Synchronization for Windows system manager is a separate Java process that:

- Leverages the product's back-end networked facilities to dynamically deliver configuration updates to connectors
- Keeps status of each connector and all connector subcomponents

■ Coordinates `idsync resync` operations that are used to initially synchronize two directories

Central Logger

Connectors may be installed so that they are widely distributed across remote geographical locations; therefore, it is of great administrative value to have all logging information centralized, which allows the administrator to monitor synchronization activity, detect errors, and evaluate the health of the entire system from a single location.

Administrators can use the central logger logs to:

■ Verify that the system is running correctly
■ Detect and resolve individual component and system-wide problems
■ Audit individual and system-wide synchronization activity
■ Track a user's password synchronization between directory environments

There are two different types of logs:

■ The **audit log** provides information about the system's day-to-day activities, which includes important events such as a user's password being synchronized between directories. You can control the level of information that is logged in the audit log by increasing or decreasing the detail provided in the log messages.

Note – Identity Synchronization for Windows also writes all of the error log messages to the audit log to facilitate easy correlation with other events.

■ The **error log** provides information about conditions qualified as severe errors and warnings. All error log entries are worthy of attention, so you cannot prevent errors from being logged. If an error condition takes place, it will always be documented in the error log.

Connectors

A connector is a Java process that manages the synchronization process in a single data source type. A connector detects user changes in the data source, and publishes these changes to remote connectors over Message Queue.

Identity Synchronization for Windows provides the following directory-specific connectors, which are responsible for bidirectionally synchronizing user attributes and password updates between directories and domains:

■ **Directory Server Connector**: Supports a single root suffix (for example suffix/database) in a Directory Server

■ **Active Directory Connector**: Supports a single instance in a Windows 2000 or Windows 2003 Server Active Directory environment. You may use multiple connectors for additional domains

- **Windows NT Connector**: Supports a single domain in a Windows NT environment

Note – The Watchdog is installed anywhere you install a connector, and it is responsible for starting, restarting, and stopping the connectors. For more information, see "Watchdog Process" on page 90.

Connector Subcomponents

A subcomponent is a lightweight process or library that runs separately from the connector. Connectors use subcomponents to access native resources that cannot be accessed remotely, such as capturing passwords inside Directory Server or Windows NT.

The following connector subcomponents are configured or installed with the directory being synchronized and communicated with the corresponding connector over an encrypted connection.

- "Directory Server Plug-in" on page 93
- "Windows NT Connector Subcomponents" on page 93

Note – Active Directory Connectors do not require subcomponents.

Directory Server Plug-in

The Directory Server Plug-in is a subcomponent of the Directory Server Connector. You configure the Directory Server Plug-in on each Directory Server being synchronized.

This Plug-in:

- Enhances the Directory Server Connector's change-detection features by storing encrypted passwords in the Retro Changelog

- Provides bidirectional support for user attribute and password synchronization between Active Directory and Directory Server (see "Using On-Demand Password Synchronization to Obtain Clear-Text Passwords" on page 100

Note – The Directory Server Plug-in is functional in a N-way, multimaster replication (MMR) environments. (Previously, Identity Synchronization for Windows supported two-way MMR only.)

Windows NT Connector Subcomponents

If your installation requires synchronization with Windows NT SAM Registries, the Identity Synchronization for Windows installation program installs the following in the Primary Domain Controller (PDC) along with the Windows NT Connector:

- **Change Detector**: Detects user entry and password change events by monitoring the Security Log, and then passes the changes to the Connector
- **Password Filter**: Captures password changes made on the NT Domain Controller and passes these securely to the NT Connector

Message Queue

Identity Synchronization for Windows uses Message Queue (a persistent message queue mechanism with a publish/subscribe model) to propagate attribute and password changes between directory sources and to distribute administrative and configuration information to the connectors managing synchronization for those directory sources.

Message Queue is an enterprise messaging system that implements the Java Message Service (JMS) open standard. The JMS specification describes a set of programming interfaces that provide a common way for Java applications to create, send, receive, and read messages in a distributed environment.

Message Queue consists of message publishers and subscribers that exchange messages using a common message service. This message service is composed of one or more dedicated message brokers, which are responsible for controlling access to the message queue, maintaining information about active publishers and subscribers, and ensuring that messages are delivered.

Message Queue is the best approach because it:

- Establishes a system of trust between connectors
- Simplifies security access controls for all components
- Facilitates end-to-end encryption of passwords
- Ensures that all password update messages are delivered
- Reduces connector-to-connector communication complexity and security risks
- Enables a central authority to distribute configuration information
- Allows for the aggregation of all connector logs in a central location

System Components Distribution

Before you can develop an effective deployment, you must understand how Identity Synchronization for Windows components are organized and how the product operates. This section is organized as follows:

When you understand the basic concepts described in this section and in the Deployment Scenario example (on "Deployment Example: A Two-Machine Configuration" on page 103), you should be able to extrapolate the information to create deployment strategies for more complex, sophisticated scenarios (such as mixed Active Directory and Windows NT environments or multi-server environments).

Core

Install all Core components only once in any of the supported operating system's directory servers. Identity Synchronization for Windows installs Administration Server on your machine if it is not already installed. Install Message Queue 3.6 Enterprise Edition on the same machine where you are planning to instal Core.

Directory Server Connector and Plug-in

You can install Directory Server Connectors on any of the supported operating systems. You are not required to install a Directory Server Connector on the same machine where the Directory Server being synchronized is running. However, there must be one Directory Server Connector installed for each configured Directory Server source.

You must configure the Directory Server Plug-in on every host where a Directory Server to be synchronized resides.

Note – A single Directory Server Connector is installed for each Directory Server source. However, Directory server Plug-ins should be configured for each master, hub, and consumer replica to be synchronized.

Active Directory Connector

You can install Active Directory Connectors on any of the supported operating systems. You are not required to install an Active Directory Connector in the Windows environment; however, there must be one Active Directory Connector installed per Active Directory domain.
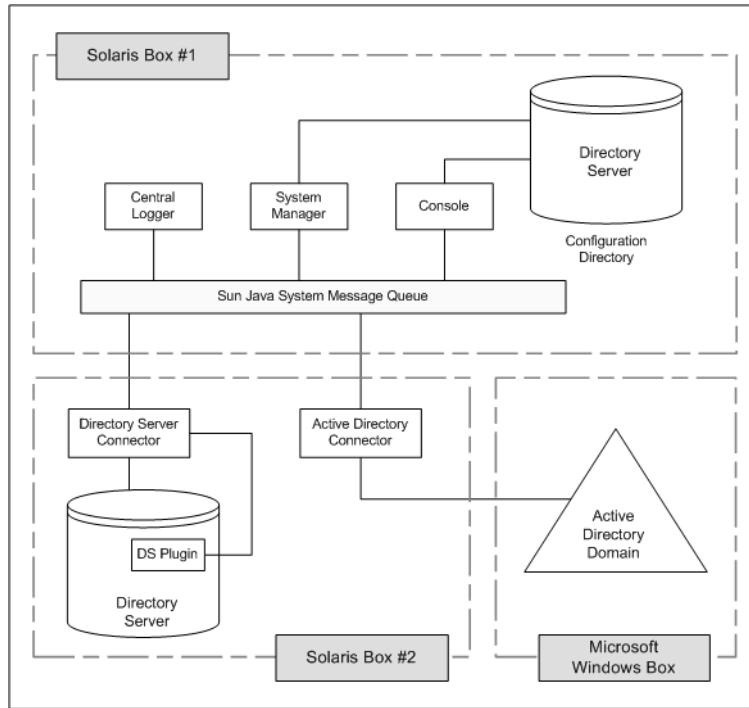
FIGURE 3–2   Directory Server and Active Directory Component Distribution

Windows NT Connector and Subcomponents

To synchronize with Windows NT SAM Registries you must install the Windows NT Connector in the Primary Domain Controller (PDC). In addition, the installation program installs the two NT Connector subcomponents (the Change Detector and the Password Filter DLL) along with the Connector in the PDC of the NT Domain. A single NT Connector synchronizes users and passwords for a single NT Domain.
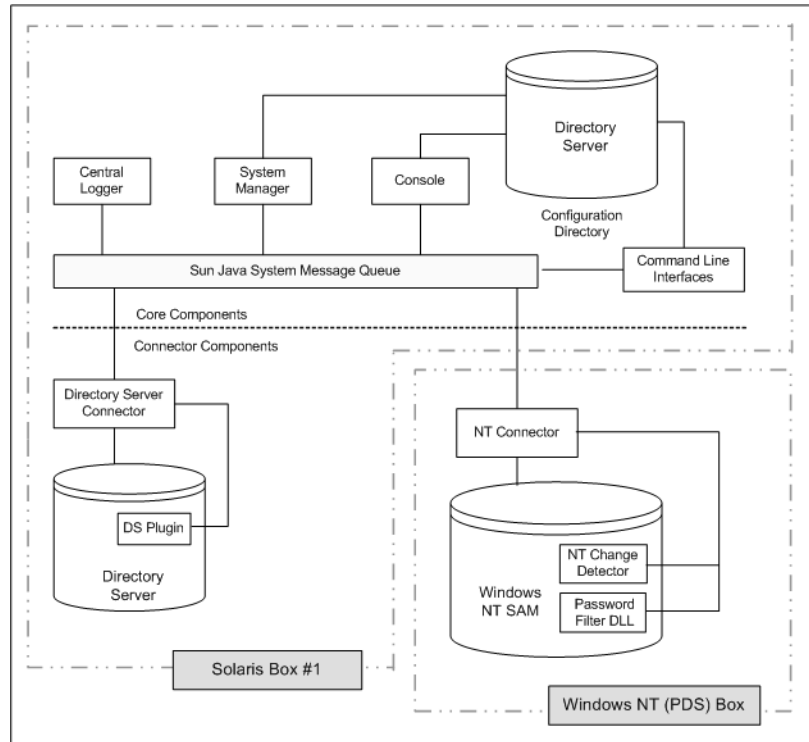
FIGURE 3–3   Directory Server and NT Component Distribution

How Identity Synchronization for Windows Detects Changes in Directory Sources

This section explains how user entry and password changes are detected by Sun Java System Directory Server (Directory Server), Windows Active Directory, and Windows NT Connectors.

The information is organized as follows:

- "How Directory Server Connectors Detect Changes" on page 97
- "How Active Directory Connectors Detect Changes" on page 98
- "How Windows NT Connectors Detect Changes" on page 99
- "Propagating Password Updates" on page 100

How Directory Server Connectors Detect Changes

The Directory Server Connector examines the Directory Server retro changelog over LDAP to detect user entry and password change events. The Directory Server Plug-in helps the Connector to do the following:

For more information about retro changelog, see "Replication and the Retro Change Log Plug-In" in *Sun Java System Directory Server Enterprise Edition 6.0 Reference* in *Sun Java System Directory Server Enterprise Edition 6.0 Reference*

- Capture clear-text passwords by encrypting and then making them available in the Retro-Changelog. Without the Plugin, only hashed passwords appear in the Retro-Changelog and hashed passwords cannot be synchronized.

- Perform On-Demand Password Synchronization with Active Directory; removing the need to install any Identity Synchronization for Windows components in a Windows environment (See "Using On-Demand Password Synchronization to Obtain Clear-Text Passwords" on page 100
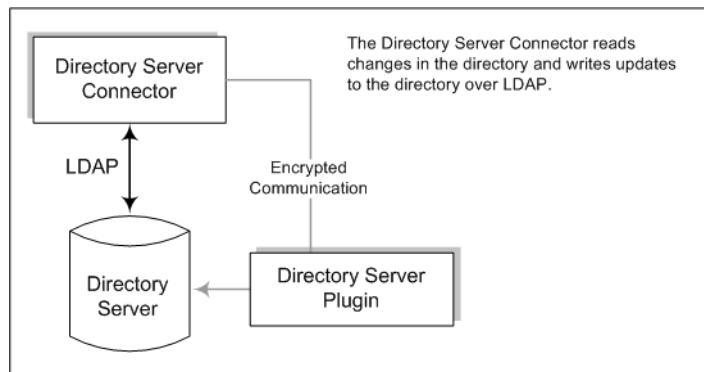
FIGURE 3–4   How Directory Server Connectors Detect Changes

How Active Directory Connectors Detect Changes

The Windows 2000/2003 Server Active Directory Connector detects user entry and password changes by examining the Active Directory `USNChanged` and `PwdLastSet` attribute values.

Unlike the Directory Server's Retro Changelog, when you change attributes in an entry, Active Directory does not report which attributes changed. Instead, Active Directory identifies entry changes by incrementing the `USNchanged` attribute. To detect changes to individual attributes, Active Directory and Windows NT Connectors use an in-process database called the *object cache*. The object cache stores a hashed copy of each Active Directory entry, which allows the Connector to determine exactly which attributes were modified in the entry.

You are not required to install Active Directory Connectors in the Windows environment. They can run also on other platforms such as Solaris or Linux and detect or make changes remotely over LDAP.
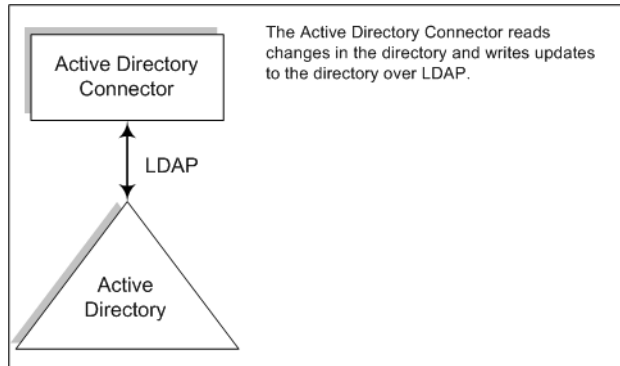
FIGURE 3–5   How Active Directory Connectors Detect Changes

How Windows NT Connectors Detect Changes

The Windows NT Connector detects user entry and password changes by examining the Security log for audit events about user objects.

To synchronize with Windows NT SAM Registries (see "Windows NT Connector and Subcomponents" on page 96) you must install the Windows NT Connector in the Primary Domain Controller (PDC). In addition, the installer installs the two NT Connector subcomponents (the Change Detector and the Password Filter DLL) along with the Connector in the PDC of the NT Domain. A single NT Connector synchronizes users and passwords for a single NT Domain.
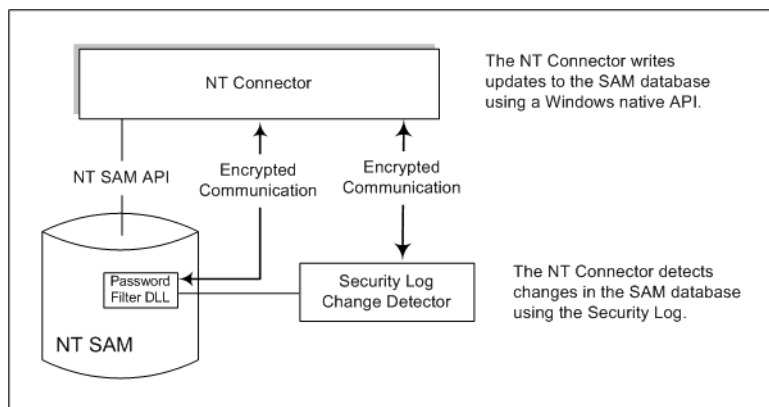
FIGURE 3–6   How Windows NT Connectors Detect Changes

> **Note** – If you have a Windows NT machine in your deployment, auditing must be enabled or Identity Synchronization for Windows cannot read log messages from that machine. To verify whether audit logging is enabled on your Windows NT machine, see "Enabling Auditing on a Windows NT Machine" on page 258
>
> For a description of the Change Detector and the Password Filter DLL subcomponents, review "Windows NT Connector Subcomponents" on page 93

Propagating Password Updates

This section explains the following methods for obtaining clear-text passwords needed to propagate password changes between Windows systems and Directory Server systems:

- "Using the Password Filter DLL to Obtain Clear-Text Passwords" on page 100
- "Using On-Demand Password Synchronization to Obtain Clear-Text Passwords" on page 100

Using the Password Filter DLL to Obtain Clear-Text Passwords

Windows NT Connectors must obtain clear-text passwords to propagate password updates to the Sun Java System Directory Server. However, you cannot extract clear-text passwords from a Windows directory. By the time passwords are stored in the directories, they have already been encrypted.

Windows NT provides a Password Filter DLL interface, which allows components to capture clear-text passwords before they are stored in a directory permanently.

Using On-Demand Password Synchronization to Obtain Clear-Text Passwords

While Active Directory supports the same password filter as Windows NT, you must install the password filter DLL on every domain controller instead of just the Primary Domain Controller (PDC). Because this can be a significant installation burden, Identity Synchronization for Windows uses a different approach, called *on-demand password synchronization*, to synchronize password changes from Active Directory to Directory Server.

On-demand password synchronization provides a method to obtain new password values on Directory Server when the user tries to log-in after their password changes on Windows 2000.

On-demand password synchronization also allows you to synchronize passwords on Active Directory without the need of a password filter DLL.

▼ The on-demand password synchronization process occurs as follows:

1 **User presses Ctrl-Alt-Del on a Windows workstation and changes his or her password. New passwords are stored in Active Directory.**

2 **The Active Directory Connector polls the system at scheduled intervals.**

When the Connector detects the password change (based on changes made to the USNchanged (Update Sequence Number) and PwdLastSet attributes), the Connector publishes a message on Message Queue about the password change. The message is transferred on an SSL-encrypted channel.
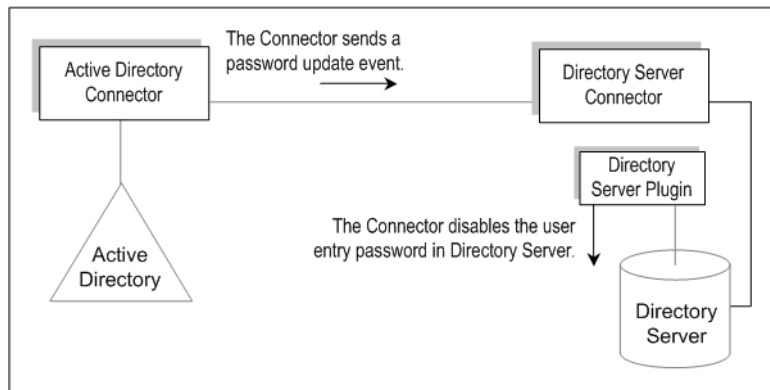
FIGURE 3–7 On-Demand Password Synchronization-Part 1

3 **The Directory Server Connector receives the password change message from Message Queue (over SSL).**

4 **The Directory Server Connector sets the user entry's dspswvalidate attribute to true which invalidates the old password and alerts the Directory Server Plug-in of the password change.**

5 **When the user tries logging on, using an LDAP application (such as Portal Server) to authenticate against the Directory Server, the Sun Java System Directory Server Plug-in detects that the password value in the Directory Server entry is invalid.**

6 **The Directory Server Plug-in searches for the corresponding user in Active Directory. When the Plug-in finds the user, the Plug-in tries to bind to Active Directory using the password provided when the user tried logging into Directory Server.**

Note – On-demand password synchronization requires the application to use simple authentication against the Directory Server instead of using a more-complex authentication mechanism, such as SASL Digest-MD5.

7   **If the bind against Active Directory succeeds, then the user provided his or her new Active Directory password and the Directory Server Plug-in set the password and removed the invalid password flag from the user entry on Directory Server.**
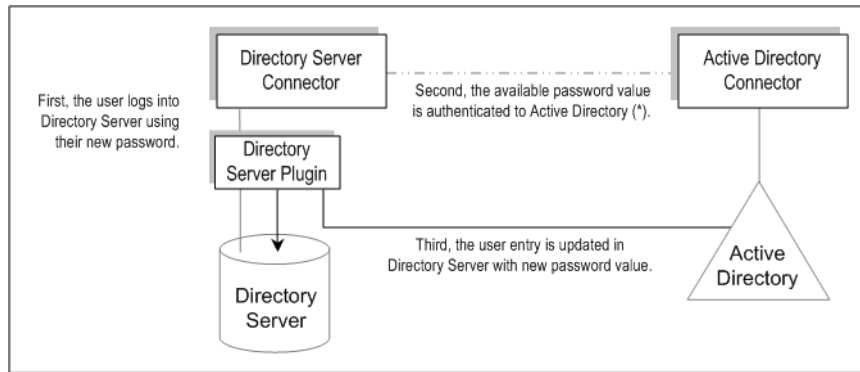
FIGURE 3–8   On-Demand Password Synchronization-Part II

Note – If the user authentication fails, the user entry password remains in Directory Server and the passwords on Directory Server and Active Directory will be out-of-sync until the user logs in with a valid password (one that authenticates to Active Directory).

Reliable Synchronization

Identity Synchronization for Windows takes many precautions to ensure that you do not lose user change events— even when components become temporarily unavailable. Identity Synchronization for Windows' reliability is similar to the TCP network protocol. TCP guarantees that even over a lossy and intermittently connected network, it will eventually deliver all data in order. Data sent during a temporary network outage is queued while the network is down and re-delivered once connectivity is restored. Identity Synchronization for Windows will eventually detect and apply user change events if one of the following components becomes temporarily unavailable:

- Connector
- Directory Server
- Message Queue
- Active Directory Domain Controller
- Windows NT Primary Domain Controller
- System Manager
- Configuration Directory

If one of these components is not available, Identity Synchronization for Windows will delay synchronization until the affected component is available without losing any changes (even to passwords). This version of Identity Synchronization for Windows does not support Sun Cluster or other true, high-availability solutions. Because Identity Synchronization for Windows is a behind-the-scenes application that users do not interact with directly, high availability is not usually required. If you ever experience a catastrophic failure, you can re-install Identity Synchronization for Windows components and use the `idsync resync` command to re-synchronize all directory sources.

In most situations, when a component is unavailable, the program queues synchronization events and applies them only when the component becomes available. There are two exceptions to this process:

- In a multi-master replication (MMR) Directory Server environment, external changes to Windows users can be synchronized to the preferred or secondary Directory Server(s).

  If the preferred Directory Server is unavailable, then the Directory Server Connector will apply changes to one of the available secondary servers from the MMR topology.

- While the Active Directory Connector can communicate with a single Active Directory domain controller only, the Directory Server Plug-in can fail between all Active Directory domain controllers while performing on-demand password synchronization. This point is where failover is most important— if the Directory Server Plug-in cannot contact an Active Directory domain controller to verify a user's new password, the user cannot log into Directory Server.

Deployment Example: A Two-Machine Configuration

This section describes a deployment scenario in which Identity Synchronization for Windows is used to synchronize user object creation and bidirectional password modification operations between Sun and Windows directories.

The deployment scenario consists of two systems:

- A system running a Sun Java System Directory Server (host name: *corp.example.com*)
- A system running Active Directory on a Windows 2000 server (host name: *sales.example.com*)

Note – Though NT is not used in this scenario, it is important to note that Identity Synchronization for Windows also supports synchronization with NT domains.

This example illustrates the synchronization requirements (node structures with associated attribute values) used for this deployment scenario.
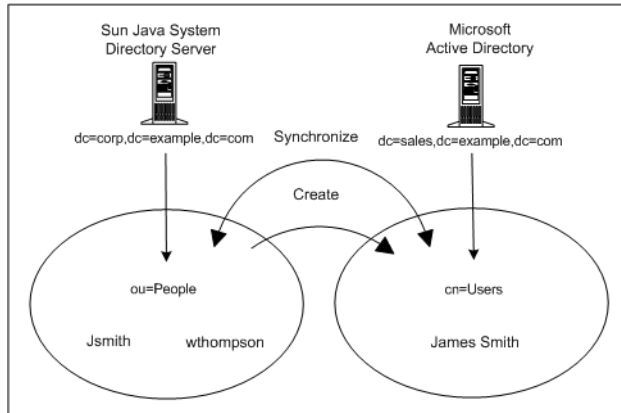
FIGURE 3–9   Synchronization Requirements

There are two goals for this scenario:

- To synchronize user passwords bidirectionally between the *user subtrees* (*ou=people* in Directory Server and *cn=users* in Active Directory), which means that whenever a user password changes in either directory, the password change is synchronized to the associated user in the other directory.

  For example, if you change the password for *uid=JSmith* in the *ou=people* container on the Directory Server, then the new password should automatically be synchronized to cn=Joe Smith in the *cn=users* container on the Active Directory server.

- To synchronize user object creation operations from the Directory Server people subtree to the Active Directory user subtree only.

  For example, if you create a new user (uid=WThompson in the *ou=People* container) with a specified set of attributes, then Identity Synchronization for Windows will create a new account for WThompson (cn=William Thompson in the *cn=Users* container) with the same set of attributes on Active Directory.

Note – Identity Synchronization for Windows supports multiple synchronization sources of the same type (for example, you can have more than one Directory Server in a deployment or multiple Active Directory domains).

Creation, modification, and deletion synchronization settings are global for the entire set of directories, and cannot be specified for individual directory sources. If you synchronize user object creations from Directory Server to Windows directories, then user object creations will propagate from *all* Directory Servers to *all* Active Directory domains and Windows NT domains configured in the installation.

Physical Deployment

This section illustrates how all the product's components are physically deployed on a single Solaris box, while the Active Directory domain resides in a separate Active Directory domain controller where no components have been installed.
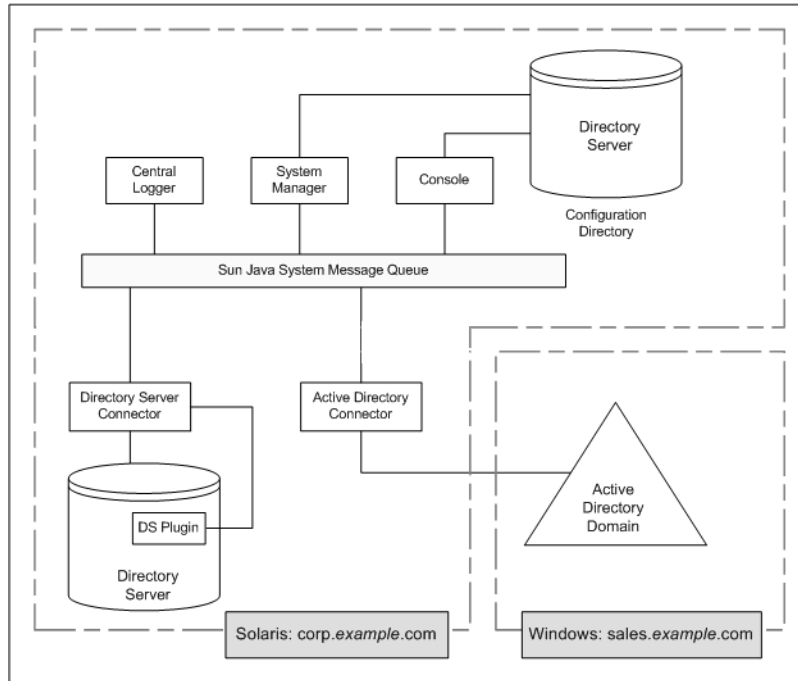
FIGURE 3–10   Directory Server and Active Directory Scenario

Component Distribution

Host *corp.example.com* is a Directory Server installed in a Solaris operating system. The root suffix for the Directory Server being synchronized is *dc=corp,dc=example,dc=com*.

This machine contains:

- Identity Synchronization for Windows Core components
- Identity Synchronization for Windows Directory Server Connector
- Identity Synchronization for Windows Directory Server Plug-in
- Identity Synchronization for Windows configuration directory (located in a different Directory Server instance than the one being synchronized)

    Host *sales.example.com* is the Active Directory domain being synchronized.

4

Preparing for Installation

Before installing Identity Synchronization for Windows 6.0 or before you migrate from Sun Java SystemIdentity Synchronization for Windows 1 2004Q3 SP1 to version 6.0, you should familiarize yourself with the installation and configuration process.

For information on the Identity Synchronization for Windows installation requirements, see Chapter 5, "Identity Synchronization for Windows Bugs Fixed and Known Problems," in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*

This chapter describes these processes and provides other information you may find helpful as you prepare to install the product. This information is organized into the following sections:

- "Installation Overview" on page 107
- "Configuration Overview" on page 112
- "Synchronizing Passwords with Active Directory" on page 116
- "Configuring Windows for SSL Operation" on page 122
- "Installation and Configuration Decisions" on page 123
- "Installation Checklists" on page 126

Installation Overview

This section illustrates a single-host installation procedure for Identity Synchronization for Windows.

```
┌─────────────────────────────────┐
│   Upgrade/Install Directory Server 6.0   │
│        and Message Queue 3.6             │
│           Enterprise Edition             │
└─────────────────────────────────┘
              ┊
  ─ ·· ─ ·· ─ ──── ─ ·· ─ ·· ─
              │
┌─────────────────────────────────┐
│ Download and Unpack/Unzip Identity Synchronization │
│          for Windows Binaries File       │
└─────────────────────────────────┘
              │
┌─────────────────────────────────┐
│      Install Core and Sun Java System     │
│          Administration Server           │
└─────────────────────────────────┘
              │
┌─────────────────────────────────┐
│        Configure Directory Sources       │
│   (includes Preparing Directory Servers)  │
└─────────────────────────────────┘
              │
┌─────────────────────────────────┐
│  Install Directory Server Connector for Every │
│  Configured Directory Server. Configure Directory │
│  Server plugin. You can choose to configure the │
│  Directory Server plug-in at a later point too, but this │
│    should be done before synchronization.  │
└─────────────────────────────────┘
              │
┌─────────────────────────────────┐
│ Install Windows (Active Directory or NT) Connector │
│      for Every Domain Being Synchronized  │
└─────────────────────────────────┘
              │
┌─────────────────────────────────┐
│   Run idsync resync Command Line Utility  │
│          to Initialize the System        │
└─────────────────────────────────┘
              │
┌─────────────────────────────────┐
│           Start Synchronization           │
└─────────────────────────────────┘
```

FIGURE 4–1   Single-Host Installation Procedure

Some components must be installed in a particular order, so be sure to read all installation instructions carefully.

Identity Synchronization for Windows provides a "To Do" list, which is displayed throughout the installation and configuration process. This information panel lists all of the steps you must follow to successfully install and configure the product.

This is a list of remaining installation and configuration steps:

✓ 1 : Install the Identity Synchronization core components.

2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importcnf'.

3 : Prepare every Sun Directory Server included in this configuration by using the console or the 'idsync prepds' command.

4 : Install connectors for every configured directory source.

5 : After installing each Sun Directory Server connector, configure the Sun Directory Server plugin on every master and on every read-only replica by using the console or the "idsync dspluginconfig" command.

6 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.

7 : Start synchronization using the console or the 'idsync startsync' command.

FIGURE 4–2   To Do List for Identity Synchronization for Windows Installation and Configuration

As you go through the installation and configuration process, the program greys-out all completed steps in the list (as you can see in "Installation Overview" on page 107).

The rest of this section provides an overview of the installation and configuration process, and is organized as follows:

- "Installing Core" on page 109
- "Configuring the Product" on page 110
- "Preparing the Directory Server" on page 110
- "Installing Connectors and Configuring Directory Server Plug-in" on page 110
- "Synchronizing Existing Users" on page 111

Installing Core

When you install Core, you will be installing the following components:

- **Sun Java System Administration Server**: Configures the Directory Server Plug-in and provides the administration framework.

- **Console**: Provides a centralized location for performing all of the product's component configuration and administration tasks

- **Central logger**: Centralizes all audit and error logging information in a central location

- **System manager**: Delivers configuration updates to connectors dynamically and maintains the status of each connector

---

Note – Instructions for installing Core are provided in Chapter 5

Configuring the Product

After installing Core, you use the Console to initially configure the directory sources to be synchronized (and other characteristics of the deployment) all from a centralized location.

Note – Instructions for configuring directory resources are provided in Chapter 6

Preparing the Directory Server

Directory Server Connectors support the Sun Java System Directory Server 6.0.

Before you can install Directory Server Connectors, you must prepare a Sun Java System Directory Server source for every configured Directory Server master (both preferred and secondary masters) being synchronized.

You can perform this task from the Console or from the command line using the `idsync prepds` subcommand.

Note – Instructions for preparing Directory Server are provided in "Preparing Sun Directory Source" on page 153.

Installing Connectors and Configuring Directory Server Plug-in

You can install any number of connectors depending on the number of configured directories available in your system. Both the Console and the installation program use the directory label to associate a connector with the directory that is synchronized.

TABLE 4–1 Label Naming Conventions

Connector Type	Directory Source Label	Subcomponent
Directory Server Connector	root suffix or suffix/database	Directory Server Plug-in
		Configure one Plug-in in every Directory Server (master or consumer) for the root suffix being synchronized.
AD Connector	Domain name	None

TABLE 4–1   Label Naming Conventions   *(Continued)*

Connector Type	Directory Source Label	Subcomponent
NT Connector	Domain name	(Automatically installed with the Window NT Connector) Change Detector and Password Filter DLL subcomponents are installed together in the same installation.
		You must install the Windows NT Connector using the graphical user interface (GUI) installer.

TABLE 4–2   Label Naming Examples

Connector Name	Directory Source
CNN100	SunDS1 on ou=isw_data1
CNN101	AD1
CNN102	SunDS1 on ou-isw_data2
CNN103	SunDS2

Note – Instructions for installing and configuring Connectors are provided in

Synchronizing Existing Users

After installing the connectors, plug-ins, and subcomponents you must run the `idsync resync` command line utility to bootstrap deployments with existing users. This command uses administrator-specified matching rules to

- Link existing entries (For more information about *linking users* , see "Linking Users" on page 213
- Populate an empty directory with the contents of a remote directory
- Bulk-synchronize attribute values (including passwords) between two existing user populations where entries in both the Windows and Directory Server directories are uniquely identified and linked to each other.

For more information about synchronizing existing users, see .

Configuration Overview

After installing the product, you must configure the product deployment, which includes:

- Configuring the directories and global catalogs to be synchronized
- Specifying synchronization settings for attribute modifications and object activations/inactivations
- Specifying settings for group synchronization
- Specifying settings for account lockout and unlockout synchronization
- Specifying synchronization settings for (optional) user entry creations and deletions between the configured directories

This section provides an overview of the following configuration element concepts:

- "Directories" on page 112
- "Synchronization Settings" on page 112
- "Objectclasses" on page 113
- "Attributes and Attribute Mapping" on page 113
- "Synchronization User Lists" on page 115

Note – For more information about configuration instructions, see Chapter 6.

Directories

A directory represents:

- A single root suffix (suffix/database) in one or more Sun Java System Directory Servers
- A single Active Directory domain in a Windows 2000 or Windows 2003 Server Active Directory forest
- A single Windows NT domain

  You can configure any number of each directory type.

Synchronization Settings

You use synchronization settings to control the direction in which object creations, object deletions, password and other attribute modifications are propagated between Directory Server and Windows directories. Synchronization flow options are as follows:

- From Directory Server to Active directory/Windows NT
- From Active directory/Windows NT to Directory Server
- Bidirectionally

Note – In a configuration that includes Active Directory and Windows NT, it is not possible to save a configuration that specifies different synchronization settings for creations or modifications between Windows NT and Directory Server, and between Active Directory and Directory Server.

Objectclasses

When you configure resources, you will specify which entries to synchronize based on their *objectclass*. Object class(es) determine which *attributes* will be available to synchronize for both Directory Server and Active Directory.

Note – Objectclasses are not applicable for Windows NT

Identity Synchronization for Windows supports two types of objectclasses:

- **Structural Objectclasses**: Every entry that's created or synchronized from the selected Directory Server must have at least one structural objectclass. Select a structural objectclass from the drop-down list. (*Defaults to inetorgperson* on Directory Server and *User* on Active Directory)
- **Auxiliary Objectclasses**:
  - **Directory Server** allows you to select one or more objectclasses from the Available Auxiliary Object Classes list pane to augment the selected structural class, which provides additional attributes for synchronization.
  - **Active Directory** is more restrictive with the auxiliary objectclass. Attributes on all valid auxiliary objectclasses for the selected structural objectclass will be available for synchronization.

Note – For detailed information about configuring objectclasses and attributes, see Chapter 6

Attributes and Attribute Mapping

Attributes hold descriptive information about a user entry. Every attribute has a label, one or more values, and follows a standard syntax for the type of information that can be stored as the attribute value(s).

Note – You can define attributes from the Console. Instructions for defining attributes are provided in Chapter 6.

Attribute Types

Identity Synchronization for Windows synchronizes *significant* and *creation* user attributes, as follows:

- **Significant Attributes**: Synchronized between Sun and Windows directories whenever the attributes are modified according to specified modification synchronization settings.

- **Creation Attributes**: Synchronized between Sun and Windows directories whenever a new user is created, according to specified object creation synchronization settings.

  Mandatory creation attributes are attributes that are considered "mandatory" in order to successfully complete a creation action in the target directory. For example, Active Directory expects that both cn and samaccountname have valid values upon creation. On the Sun side, if you are configuring inetorgperson of a user objectclass, Identity Synchronization for Windows will expect cn and sn as mandatory attributes for a creation.

  A creation attribute default updates the target directory creation attribute with a default value *only* when there is no value in the attribute propagated from the originating directory. (Creation attribute defaults can be based on other attribute values. See "Parameterized Attribute Default Values" on page 114.)

Note – Significant attributes are automatically synchronized as creation attributes but not the other way around. Creation attributes are only synchronized during user creations.

Parameterized Attribute Default Values

Identity Synchronization for Windows allows you to create *parameterized* default values for creation attributes using other creation or significant attributes.

To create a parameterized default attribute value, you embed an existing creation or significant attribute name — preceded and followed by percent symbols (%*attribute_name*%) — in an expression string. For example, homedir=/home/%uid% or cn=%givenName%. %sn%.

When you create these attribute default values:

- You can use multiple attributes in a creation expression (cn=%givenName% %sn%), but the attributes in % *attribute_name*% must have single values.

- If A=0, then B can have one default value only.

- You can use the backslash symbol (\\) for quoting (for example, diskUsage=0\\%).

- Do not use expressions that have cyclic substitution conditions (for example, sn=%uid% and uid= %sn%.

Mapping Attributes

After you define the attributes, map the attribute names between the Directory Server and Active directory/Windows NT systems to synchronize them to each other. For example, you must map the Sun `inetorgperson` attribute to the Active Directory `user` attribute.

Note – You use attribute maps for both significant and creation attributes, and you must configure attribute maps for all "mandatory creation attributes" in each directory type.

Synchronization User Lists

You create Synchronization User Lists (SULs) to define specific users in both the Sun and Windows directories to be synchronized. These definitions enable synchronization of a flat Directory Information Tree (DIT) to a hierarchical directory tree.

The following concepts are used to define a Synchronization User List:

- **Base DN** (not applicable to Windows NT): Includes all users in that DN unless another SUL is more specific or unless excluded by a filter.
- **Filter**: Uses attributes in the user's entry to exclude users from synchronization or to separate users with the same base DN into multiple SULs. This filter uses LDAP filter syntax.
- **Creation expression** (not applicable to Windows NT): Constructs the DN where new users are created, for example, `cn=%cn%,ou=sales,dc=example, dc=com` where `%cn%` is replaced with the value of `cn` from the existing user entry. A creation expression must end with the base DN.

  An SUL includes two definitions; where each definition identifies the group of users to be synchronized in the topology terms of the directory type.

- One definition identifies which Directory Server users to synchronize (for example: `ou=people, dc=example, dc=com`)
- The other definition identifies the Windows users to synchronize (for example: `cn=users, dc=example, dc=com`)

  When you are preparing to create SULs, ask yourself the following questions:

- Which users will be synchronized?
- Which users are excluded from synchronization?
- Where should new users be created?

Note – See Appendix D for detailed information about creating SULs.

Synchronizing Passwords with Active Directory

The default password policy on Windows 2000 was changed on Windows 2003 to enforce strict passwords by default.

Identity Synchronization for Windows services must occasionally create entries that do not have passwords (for example, during a resync -c from Directory Server to Active Directory). Consequently, if you have password policies enabled on Active Directory (on Windows 2000 or 2003) or on Directory Server, user creation errors can result.

Although you do not have to disable password policies on Active Directory or Directory Server, you should understand the issues associated with enforcing password policies on the different systems.

The following installation information is important if you will be synchronizing passwords with Active Directory on Windows 2003 Server Standard or Enterprise Edition:

- If you are installing on Windows, you can install the Active Directory Connector on Solaris, Linux, or Windows.

  Note – Active Directory Connectors will work with Active Directory on both Windows 2000 and Windows 2003 Server.

- You use the same procedures to create directory sources, global catalogs, and Synchronization User Lists for Windows 2003 Server that you used for Active Directory on Windows 2000.
- On Windows 2003 Server, the default password policy enforces strict passwords, which is not the default password policy on Windows 2000.

Enforcing Password Policies

This section explains how the password policies for Active Directory on Windows 2003 Server, Windows 2000, and Sun Java System Directory Server 6.0 can affect synchronization results.

The information is organized as follows:

Overview

If you create users on Active Directory (or Directory Server) that meet the required password policies for that system, the users may be created and synchronized properly between the two systems. If you have password policies enabled on both systems, the passwords must meet the policies of both systems or the synchronized user creations will fail.

- If you enable the password policy features on Active Directory, you should enable a similarly configured or matched password policy on Directory Server.

- If you cannot create a consistent password policy on both Active Directory and Directory Server, you should enable password policies on the side that you consider the authoritative source for passwords and user creations. However, there are some cases in which user creations will not work as expected because of certain password policy configurations.

Important Notes

The following sections provide important information about password policies:

Directory Server Password Policies

If you create users in Active Directory with passwords that violate the Directory Server password policy, those users will be created and synchronized in Directory Server, but the entries will be created without a password. The password will not be set until the new user logs into Directory Server, which triggers on-demand password synchronization. At this time the login will fail because the password violates the Directory Server password policy.

There are several ways to recover from this situation:

- Force the user to change their password the next time they log on to Active Directory

- Change the user password on Active Directory, and be sure the new password meets Directory Server password policy requirements

  You may want to review whether the password policy set on Active Directory and on Directory Server are equivalent (or as similar as possible).

Active Directory Password Policies

If you create users on Active Directory that do not match the Active Directory password policy, those users *will* be created on Directory Server.

- Active Directory actually creates users "temporarily" and then deletes the entries if the password does not meet the password policy requirements. Consequently, the Active Directory Connector sees this temporary ADD and creates users on the Directory Server side.

The users will not have a password in Directory Server, so no one will be able to log in as the user. In addition, these entries will not be linked to a valid entry in Active Directory. If deletions are synchronized from Active Directory to Directory Server, then the temporarily created users will be deleted automatically.

- Users are created without a password on Directory Server. Directory Server does not enforce the password policy for user creations unless the entries contain a password.

  There are several ways to recover from this situation. The preferred method is to synchronize deletions from Active Directory to Directory Server. Alternatively, you can remove the user from Directory Server and then add them to Active Directory with a valid password for the Active Directory password policies. This method ensures that the users are created on Directory Server and linked properly. Users on Directory Server will have their password invalidated when they log into Active Directory for the first time and change their passwords.

- If you do not delete the user from Directory Server, and then try to add the Active Directory user again with a new password, the ADD to Directory Server will fail because the user already exists on Directory Server. The entries will not be linked together and you will have to run a idsync resync command to link the two separate accounts.

- If you run the idsync resync command, you must be sure to reset the passwords for the accounts on Active Directory that were linked to entries on Directory Server. Resetting the passwords invalidates those passwords on Directory Server, which then forces on-demand synchronization to update the Directory Server password the next time the user authenticates to Directory Server with their new Active Directory password.

Creating Accounts Without Passwords

In certain circumstances, such as resynchronization, Identity Synchronization for Windows must create accounts without passwords.

Directory Server

WhenIdentity Synchronization for Windows creates entries in the Directory Server, without a password, it sets the userpassword attribute to {PSWSYNC}*INVALID*PASSWORD*. The user will not be able to log into Directory Server until you reset the password. One exception to this is when you run resync with the -i NEW_USERS or NEW_LINKED_USERS option. In this case, resync will invalidate the new user's password triggering on-demand password synchronization the next time the user logs in.

Active Directory

When Identity Synchronization for Windows creates entries in the Active Directory, without a password, it sets the user's password to a randomly chosen, strong password that meets Active Directory password policy requirements. In this case, a warning message is logged and the user will not be able to log into Active Directory until you reset the password.

The following tables describe some different scenarios you might encounter as you work with Identity Synchronization for Windows:

This section describes how password policies affect synchronization and resynchronization.

Use this information as a guideline to help ensure that passwords will remain synchronized. (These tables do not attempt to describe all possible configuration scenarios because system configurations differ.)

TABLE 4–3   How Password Policies Affect Synchronization Behavior

Scenario			Results		
User Originally Created In	User Meets Password Policy In		User Created In		
	Directory Server	Active Directory	Directory Server	Active Directory	Comments
Active Directory	Yes	Yes	Yes	Yes	
	Yes	No	**Yes (see Comments)**	No	Users will be created in Directory Server. However, if deletes are synchronized from Active Directory to Directory Server then this user will be deleted immediately. See "Active Directory Password Policies" on page 117 information.
	No	Yes	**Yes**	Yes	See "Important Notes" on page 117 for more information.

TABLE 4–3 How Password Policies Affect Synchronization Behavior *(Continued)*

Scenario			Results		
User Originally Created In	User Meets Password Policy In		User Created In		
	Directory Server	Active Directory	Directory Server	Active Directory	Comments
	No	No	**Yes (see Comments)**	No	Users are created in Directory Server. However, if deletes are synchronized from Active Directory to Directory Server then this user will be deleted immediately. See "Active Directory Password Policies" on page 117 information.
Directory Server	Yes	Yes	Yes	Yes	
	Yes	No	Yes	No	
	No	Yes	No	No	
	No	No	No	No	

TABLE 4–4 How Password Policies Affect Resynchronization Behavior

Scenario			
Resync Command	User Meets Password Policy In		
	Directory Server	Active Directory	Result
`resync -c -o Sun`	N/A	Yes	User will be created in Active Directory but will not be able to log in. See "Creating Accounts Without Passwords" on page 118
	N/A	No	User will be created in Active Directory but will not be able to log in. See "Creating Accounts Without Passwords" on page 118

TABLE 4–4 How Password Policies Affect Resynchronization Behavior *(Continued)*

Scenario			
Resync Command	**User Meets Password Policy In**		
	Directory Server	**Active Directory**	**Result**
resync -c -i NEW_USERS | NEW_LINKED_USERS	Yes	N/A	User will be created in Directory Server and their password will be set when the user first logs in. See "Creating Accounts Without Passwords" on page 118
	No	N/A	User will be created in Directory Server but they cannot log in because their password violates the Directory Server password policy. See "Important Notes" on page 117 and "Creating Accounts Without Passwords" on page 118 more information.
resync -c	Yes	N/A	User will be created in Directory Server but they cannot log on until a new password value is set in Active Directory or Directory Server. See "Creating Accounts Without Passwords" on page 118
	No	N/A	User will be created in Directory Server but they cannot log on until a new password value is set in Active Directory or Directory Server. See "Creating Accounts Without Passwords" on page 118

Example Password Policies

This section describes different scenarios for Active Directory and Directory Server password policy examples using the following specifications:

For Active Directory:

- Enforce Password History: 20 days
- Max Password Age: 30 days
- Min Password Age: 0 days
- Min Password Length: 7 characters
- Passwords must meet complexity requirements: Enabled

For Directory Server:

- User must change password after reset
- User May Change Password

- Keep 20 passwords in history
- Password expires in 30 days
- Send warning 5 days before password expires
- Check password syntax: Password min length is 7 characters

Error Messages

Check the central logger `audit.log` file on the Core system for the following error message:

```
Unable to update password on DS due to password policy during on-demand synchronization:

    WARNING 125 CNN100 hostname "DS Plugin (SUBC100):
    unable to update password of entry 'cn=John Doe,ou=people,o=sun',
    reason: possible conflict with local password policy"
```

Note – For more information about password policies for Windows 2003, see
http://www.microsoft.com/resources/documentation/WindowsServ/2003/

For more information about Directory Server password policies, see Chapter 7, "Directory
Server Password Policy," in *Sun Java System Directory Server Enterprise Edition 6.0
Administration Guide*

Configuring Windows for SSL Operation

If you are planning to propagate password changes from Directory Server to Windows Active
Directory servers you must configure each Active Directory server to use SSL and install the
high-encryption pack.

The Identity Synchronization for Windows Active Directory Connector installer can
automatically set-up SSL in the Active Directory Connector if you enable LDAP over SSL in
Active Directory by automatically obtaining a certificate from a Microsoft Certificate Services
Enterprise Root certificate authority as described in:

http://support.microsoft.com/default.aspx?scid=kb;en-us;q247078

However, LDAP over SSL can more easily be configured as described in this MSDN tech note:

http://support.microsoft.com/default.aspx?scid=kb;en-us;321051

In this case, if you decided to require trusted certificates for SSL communication, you must
manually install the certificate in the Connector's certificate database as described in "Enabling
SSL in the Active Directory Connector" on page 243.

Installation and Configuration Decisions

This section gives installation and configuration summaries and details the choices you make in deploying Identity Synchronization for Windows. Have this information available before you begin the installation process. This section contains:

Core Installation

You must provide the following information when you install Core:

- **Configuration Directory Host and Port**: Specify the configuration directory host and port for the Directory Server instance on which Identity Synchronization for Windows configuration information will be stored.

  You can specify an SSL port as the configuration directory port; and if you do, you must identify the port as an SSL port during the installation process.

- **Root suffix**: Specify the root suffix for the configuration directory. All configuration information is stored under this suffix.

- **Administrator's name and password**: Specify credentials for accessing the configuration Directory Server.

- **Configuration password**: Specify a secure password to protect sensitive configuration information.

- **File system directory**: Specify the location in which to install Identity Synchronization for Windows. You must install Core in the same directory as a Directory Server Administration Server.

- **Unused port number**: Specify an available port number for the Message Queue instance.

- **Administration Server**: Specify administration server administrator's user name and password if it already exists on Directory Server.

Core Configuration

You must provide the following information when you configure Core:

- **Sun Java System Directory schema**: Specify the Directory Server data you want loaded from the configuration directory.

- **User object class (for Directory Server only)** : Specify the user object class that will be used to determine user types. Identity Synchronization for Windows derives a list of attributes (including password attributes) based on this object class. This list is populated from the schema.

- **Synchronized Attributes**: Specify user entry attributes to be synchronized between the Directory Server and the Windows environment.

- **Modifications, Creations, and Deletions flow** : Specify how you want modifications, creations, and deletions to be propagated between the Sun and Windows systems. Your options are:
  - From Directory Server to Active directory/Windows NT
  - From Active directory/Windows NT to Directory Server
  - Bidirectionally

    Specify whether to synchronize object activations and inactivations are propagated between Sun and Windows systems, and specify a method for synchronizing these objects.

- **Global Catalogs**: Specify global catalogs (repositories for Active Directory topological and schema information).

- **Active Directory schema controller**: Specify the Fully Qualified Domain Name (FQDN) of the Active Directory schema source to be retrieved from the Windows global catalog.

- **Configuration Directory**: Specify the Directory Server storing the Identity Synchronization for Windows configuration.

- **Active Directory Source**: Specify the sources used to synchronize Active Directory domains.

- **Windows NT Primary Domain Controller**: Specify the Windows NT domains to be synchronized and the name of the Primary Domain Controller for each domain.

- **Synchronization User Lists**: Use LDAP DIT and filter information to specify the users to be synchronized on Directory Server, Active Directory, and NT.

- **Sun Java System Directory Servers**: Specify Directory Server instances that store users to be synchronized.

Connector Installation and Configuring the Directory Server Plug-in

You must provide the following information when you install the connectors and the Directory Server Plug-in:

- **Configuration Directory Host and Port**: Specify the configuration directory host and port for the Directory Server instance on which Identity Synchronization for Windows configuration information will be stored.

- **Root suffix**: Specify the root suffix for the configuration directory. Use the root suffix specified during Core installation.

- **Administrator's name and password**: Specify credentials for accessing the configuration Directory Server.

- **Configuration password**: Specify a secure password to protect sensitive configuration information.

- **File system directory**: Specify the location in which to install Identity Synchronization for Windows. All components installed on the same machine must have the same installation path.

- **Directory sources**: Specify the directory source for which you want to install the connector or plug-in.

When you are installing Directory Server and Windows NT Connectors, you must specify an unused port.

When you are installing the Directory Server Connector and Plug-in, you must specify the host, port, and credentials for the Directory Server that corresponds to that Connector and Plugin.

Using the Command Line Utilities

Identity Synchronization for Windows enables you to perform a variety of tasks from the command line using the idsync script with the following subcommands:

- certinfo: Displays certificate information based on your configuration and SSL settings

- changepw: Changes the Identity Synchronization for Windows configuration password

- prepds: Prepares a Sun Java System Directory Server source for use by Identity Synchronization for Windows

- printstat: Prints the status of installed connectors, the system manager, and Message Queue

  You can also use the printstat command to display a list of the remaining installation and configuration steps you have to perform to complete the installation process.

- resetconn: Resets connector states in the configuration directory to *uninstalled* (in cases of hardware or uninstaller failure only)

- resync: Resynchronizes and links existing users, and pre-populates directories as part of the installation process

- dspluginconfig: Configures or unconfigures the Directory Server plugin.

- groupsync: Enables or disables group synchronization

- accountlockout: Enables or disables account lockout feature.

- startsync: Starts synchronization

- stopsync: Stops synchronization

Note – See Appendix A for detailed information about these utilities.

Installation Checklists

These checklists are intended to aid in the installation process. Print them out and record the following information prior to installing Identity Synchronization for Windows.

TABLE 4–5   Core Installation Checklist

Required Information	Entry
Configuration directory host and port	
Root suffix for the configuration directory (such as dc=example,dc=com)	
File system directory in which to install Identity Synchronization for Windows	
Configuration directory server administrator's name and password	
Secure configuration password to protect sensitive configuration information	
Port number for the Message Queue instance	
User name and password for the Administration Server	

TABLE 4–6   Core Configuration Checklist

Required Information	Entry
Active Directory Global Catalog (when appropriate)	
Directory Server schema server	
Directory Server User structural and auxiliary object class(es)	
Synchronized attributes	
Flow for user entry creations	
Flow for user entry modifications	
Flow for user entry activations and inactivations	
Flow for user entry deletions	
Sun Java System Directory Server directory sources	
Active Directory directory sources	
Synchronization User Lists	

TABLE 4–6 Core Configuration Checklist   *(Continued)*

Required Information	Entry
Windows source filter creation expression	
Sun Java System source filter creation expression	
User name and password for the Administration Server	
User name and password for the Administration Server	

Connector and Directory Server Plug-in Installation Checklist

Required Information	Entry
Configuration directory host and port	
Root suffix for the configuration directory	
File system directory in which to install the connector	
Configuration Directory Server administrator's name and password	
A secure configuration password to protect sensitive configuration information	
Directory sources	
An unused port for Directory Server and Windows NT	
Host, port, and credentials for the Directory Server corresponding to the Connector and Plug-in	

Linking Users Checklist

Required Information	Entry
Synchronization User Lists to be linked.	
Attributes used to match equivalent users	
XML configuration file	

Resynchronization Checklist

Required Information	Entry
Synchronization User List selection	

Required Information	Entry
Synchronization source.	
Create a user entry automatically if a corresponding user is not found at the destination directory source?	
Invalidate Directory Server passwords?	
Synchronize only those users that match the specified LDAP filter and are in the selected SULs?	

5

Installing Core

This chapter explains how to use the Identity Synchronization for Windows installation program and how to install the Identity Synchronization for Windows Core component.

The information is organized into the following sections:

Before You Begin

Before starting the Identity Synchronization for Windows installation process:

- Read Chapter 4 that contains important information, such as installation prerequisites, checklists, and administrator privilege requirements.

- A Java Runtime Environment (JRE) is not provided with this product. If necessary, you can download a Java Development Kit from the following location:

  http://java.sun.com or http://www.java.com

  You must install JRE 1.5.0_09 or later to run the Identity Synchronization for Windows installation program on your Solaris, Linux, or Windows 2000/2003 systems.

  Note – If Directory Server 6.0 is installed with Java ES, JRE 1.5.0_09 is already installed for you.

- **On Windows systems only**: You must close any open Service Control Panel windows before starting Core installation, or the installation will fail.

- **On Solaris systems**: Do not install Message Queue and Identity Synchronization for Windows in the same directory.

■ **On Red Hat Linux systems**: Do not install Message Queue and Identity Synchronization for Windows in the same directory.

Starting the Installation Program

This section explains how to download, unpack (or unzip), and run the Identity Synchronization for Windows installation program on the following platforms:

On Solaris SPARC

Use the following steps to prepare and run the Identity Synchronization for Windows installation program on a Solaris SPARC operating system.

▼ To Run Identity Synchronization for Windows on Solaris SPARC

1 **Log in as root.**

2 **Change to the directory on the delivery media for Solaris SPARC containing the installation program,** DSEE_Identity_Synchronization_for_Windows**.**

3 **Type** ./runInstaller.sh **to execute the installation program.**

To run the installation program in text-based mode, type the following.

```
./runInstaller.sh -nodisplay
```

When you run the runInstaller.sh program, Identity Synchronization for Windows automatically masks passwords so they will not be echoed in the clear.

On Solaris x86

▼ To Prepare and Run Identity Synchronization for Windows on Solaris x86

1 **Log in as root.**

2 **Change to the directory on the delivery media for Solaris x86 containing the installation program,** DSEE_Identity_Synchronization_for_Windows**.**

3 Type `./runInstaller.sh` to execute the installation program.

To run the installation program in text-based mode, type the following.

```
./runInstaller.sh -nodisplay
```

When you run the `runInstaller.sh` program, Identity Synchronization for Windows automatically masks passwords so they will not be echoed in the clear.

On Windows

Use the following steps to prepare and run the Identity Synchronization for Windows installation program on a Windows operating system:

▼ To Run Identity Synchronization for Windows on Windows

1 Log in as an Administrator.

2 Change to the directory on the delivery media for Windows containing the installation program, `DSEE_Identity_Synchronization_for_Windows`**.**

3 Type `setup.exe` to execute the installation program.

The Identity Synchronization for Windows installation wizard is displayed.

Note – Installing Core in the Administration Server root, makes the Identity Synchronization for Windows wizard detect most of the information required for installation, such as directory paths and names, and complete certain fields in the wizard panels automatically.

If any of the information is missing or incorrect, you can enter the required information manually.

Continue to the next section for Core installation instructions.

On Red Hat Linux

Use the following steps to prepare and run the Identity Synchronization for Windows installation program on a Red Hat Linux operating system:

▼ To Prepare and Run Identity Synchronization for Windows on Linux

1 Log in as root.

2 Change to the directory on the delivery media for Red Hat containing the installation program, `DSEE_Identity_Synchronization_for_Windows`**.**

3 Type `./runInstaller.sh` to execute the installation program.

To run the installation program in text-based mode, type the following.

`./runInstaller.sh -nodisplay`

When you run the `runInstaller.sh` program, Identity Synchronization for Windows automatically masks passwords so they will not be echoed in the clear.

Installing Core

This section explains the process for installing the Identity Synchronization for Windows Core on Solaris, Linux, and Windows operating systems.

Before you install Core, you should be aware of the following requirements:

- **On Solaris systems**: You must have root privileges to install and run Solaris services.
- **On Red Hat Linux systems**: You must have root privileges to install and run Linux services.
- **On Windows 2000/2003 systems**: You must have Administrator privileges to install Identity Synchronization for Windows.

Note – You must install the program as root, but after installation you can configure the software to run Solaris and Linux services as a non-root user. (See Appendix B)

You must install Core into a directory that has an existing server root managed by an Administration Server (version 5 2004Q2 or higher) or the installation program will fail. (You can install Administration Server using the Directory Server 5 2004Q2 installation program.)

Note – With Identity Synchronization for Windows 6.0, the installer checks for an existing Sun Java System Administration Server. If it is not installed, the installer will install Sun Java System Administration Server as a part of Core installation.

▼ To Install Identity Synchronization for Windows Core Components Using the Installation Wizard

1 When the Welcome screen is displayed, read the information provided and then click Next to proceed to the Software License Agreement panel.

2 Read the license agreement, then select

- **Yes (Accept License)** to accept the license terms and go to the next panel.

- **No** to stop the setup process and exit the installation program.

3   **The Configuration Location panel is displayed, specify the configuration directory location.**

FIGURE 5–1   Specifying the Configuration Directory Location

Provide the following information:

- **Configuration Directory Host**: Enter the fully qualified domain name (FQDN) of a Sun Java System Directory Server instance (affiliated with the local Administration Server) where Identity Synchronization for Windows configuration information will be stored.

  You can specify an instance on the local machine or an instance that is running on a different machine.

  Identity Synchronization for Windows allows Administrator Server to access the remotely installed instance of Directory Server.

Note – To avoid warnings about invalid credentials or host names, be sure to specify a host name that is DNS-resolvable to the machine on which the installation program is running.

- **Configuration Directory Port**: Specify the port where the configuration directory is installed. (*Default port is 389.*)

  To enable secure communication, enable the Secure Port option and specify an SSL port. (*Default SSL port is 636*).

  Once the program determines that the configuration directory is SSL-enabled, all Identity Synchronization for Windows components will use SSL to communicate with the configuration directory.

> **Note –** Identity Synchronization for Windows encrypts sensitive configuration information before sending it to the configuration Directory Server.
>
> However, if you want additional transport encryption between the Console and configuration directory, be sure to enable SSL for both Administration Server and the configuration Directory Server. Then, configure a secure connection between the Administration Server to which you will be authenticating the Directory Server Console. (For information, see the *Sun Java System Administration Server 5 2004Q2 Administration Guide*).
>
> Sun Java System Administration Server installed (and configured) as a part of the core components, is installed in a non-SSL mode.

- **Configuration Root Suffix**: Select a root suffix from the menu in which to store the Identity Synchronization for Windows configuration.

> **Note –** If the program could not detect a root suffix, and you have to enter the information manually (or if you change the default value), you must click Refresh to regenerate a list of root suffixes. You must specify a root suffix that exists on the configuration Directory Server.

4   Click Next to open the Configuration Directory Credentials panel.

FIGURE 5–2   Specifying the Administrator Credentials

5   Enter the configuration directory Administrator's user ID and password.

- If you specify admin as the user ID, you will not be required to specify the User ID as a DN.
- If you use any other user ID, then you must specify the ID as a full DN. For example, *cn=Directory Manager*.

> **Note –** If you are not using SSL to communicate with the configuration directory (see "Installing Core" on page 132), these credentials will be sent without encryption.

6   When you are finished, click Next to open the Configuration Password panel.

FIGURE 5–3 Specifying a Configuration Password

7 You must enter and confirm a password that will be used to encrypt sensitive configuration information, such as credentials. When you are done, click Next.

Note – Be sure to remember this password as it will be required whenever you want to

- Access the Identity Synchronization for Windows Console
- Create or edit a configuration
- Install components
- Run any of the command line utilities

For information about changing the configuration password see "Using changepw" on page 290.

The Select Java Home panel is displayed (see "Installing Core" on page 132). The program automatically inserts the location of the Java Virtual Machine directory to be used by the installed components.

---

8 Verify the Java Home Directory (must be a JDK/JRE 1.5.0_09 or later):

FIGURE 5–4 Specifying the Java Home Directory

- If the location is satisfactory, click Next to proceed to the Select Installation Directories panel ("Installing Core" on page 132).
- If the location is not correct, click Browse to search for and select a directory where Java is installed, for example:
- **On Solaris** : /var/java
- **On Linux**: /usr/bin/java

- **On Windows**: `C:\Program Files\j2sdk1.5`

FIGURE 5–5   Specifying the Installation Directories

9   **Enter the following information in the text fields provided or click Browse to search for and select available directories:**

- **Server Root Directory**: Specify the path and directory name of the Administration Server installation server root. The Console will be installed in this location.

- **Installation Directory** (*available only when you are installing Core on Solaris or Linux*): Specify the path and directory name of the installation directory. Core binaries, libraries, and executable will be installed in this directory.

- **Instance Directory** (*available only when you are installing Core on Solaris or Linux*): Specify the path and directory name of the instance directory. Configuration information that changes (such as log files) will be stored in this directory.

Note – There is only one server root directory available on Windows operating systems, and all products will be installed in that location.

Note – If an Administration Server corresponding to the Configuration Directory Host and Port number provided in step 3 is not found, the installer Administration Server will install the Administration Server as part of the core installation. The default port number for the Administration Server port assigned would be the configuration directory port plus one.

---

10   Click Next to proceed to the Message Queue Configuration panel.

Note – You should have installed Message Queue 3.6 Enterprise Edition before starting the Identity Synchronization for Windows installation.

On Solaris systems: Do not install Message Queue and Identity Synchronization for Windows in the same directory.

On Linux system: Do not install Message Queue and Identity Synchronization for Windows in the same directory.

On Windows systems: You must close any open Service Control Panel windows before continuing, or the Core installation will fail.

---

11   Enter the following information in the text fields provided or click Browse to search for and

Core Install: Message Queue Configuration

This product requires the use of a pre-existing Message Queue. Specify the installation location, as well as the fully qualified host name and port of the new Broker instance.

Installation Directory:	/usr	Browse...
Configuration Directory:	/var/imq	Browse...
Fully Qualified Local Host Name:	machine1.example.com	
Broker Port Number:	7676	

FIGURE 5–6   Configuring Message Queue

select available directories:

- **Installation Directory**: Specify the path of the Message Queue installation directory.
- **Configuration Directory**: Specify the path and directory name of the Message Queue instance directory.
- **Fully Qualified Local Host Name** : Specify the fully qualified domain name (FQDN) of the local host machine. (There can only be one Message Queue broker instance running per host.)

- **Broker Port Number** : Specify an unused port number for the Message Queue broker to use. (*Default port is 7676.*)

12 **Click Next and the Ready to Install panel is displayed.**

This panel provides information about the install, such as the directory where Core will be installed and how much space is required to install Core.

- If the displayed information is satisfactory, click Install Now to install the Core component (where the installation program installs the binaries, files, and packages).

- If the information is not correct, click Back to make changes.

  An "Installing" message is displayed briefly, and then the Component Configuration panel is displayed while the installation program adds configuration data to the specified configuration Directory Server. This operation includes:

  - Creating a Message Queue broker instance
  - Uploading the schema to the configuration directory
  - Uploading deployment-specific configuration information to the configuration directory

    This operation will take several minutes and may pause periodically, so do not be concerned unless the process exceeds ten minutes. (Watch the progress bar to monitor the installation program's status.)

13 **When the component configuration operation is complete, the Installation Summary panel is displayed to confirm that Identity Synchronization for Windows installed successfully.**

You can click the Details button to see a list of the files that have been installed, and where they are located.

14 **Click Next and the program will determine the remaining steps you must perform to successfully install and configure Identity Synchronization for Windows.**

A "Loading..." message, and then a Remaining Installation Steps panel each display briefly, and then the following panel ("Installation Overview" on page 107) is displayed. This panel contains a "To Do" list of the remaining installation and configuration steps. (You also can access this panel from the Console's Status tab.)

This is a list of remaining installation and configuration steps:

✓ 1 : Install the Identity Synchronization core components.

    2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importcnf'.

    3 : Prepare every Sun Directory Server included in this configuration by using the console or the 'idsync prepds' command.

    4 : Install connectors for every configured directory source.

    5 : After installing each Sun Directory Server connector, configure the Sun Directory Server plugin on every master and on every read-only replica by using the console or the "idsync dspluginconfig" command.

    6 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.

    7 : Start synchronization using the console or the 'idsync startsync' command.

FIGURE 5–7   To Do List for Identity Synchronization for Windows Installation and Configuration

The "To Do" panel will re-display throughout the installation and configuration process. The program greys-out all completed steps in the list.

Up to this point, the To Do list will contain a generic list of steps. After you save a configuration, the program provides a list of steps that are customized for your deployment (for example, which connectors you must install).

15   **After reading the list of steps, click Next and the Start Console Option panel is displayed to indicate you have finished the Core installation.**

Core Install: Start Console Option

Core installation is now complete. You must now create a configuration using the Sun Java(TM) System Console or the 'idsync importcnf' command line utility.

☑ Start the Sun Java(TM) System Console.
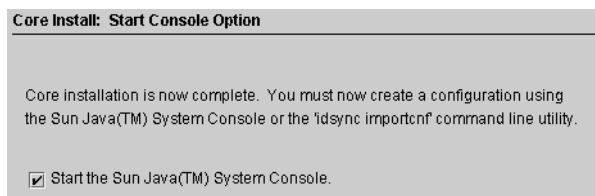
FIGURE 5–8   Starting the Console

16   **Next, you must configure the Core component, which you can do from the Sun Java System Console (*the* Start the Sun Java System Console option is enabled by default).**

If you are migrating from Identity Synchronization for Windows version 1.0 or SP1 to Sun Java System Identity Synchronization for Windows 6.0, you can import an exported version 1.0 or SP1 configuration XML document using the idsync importcnf command line utility.

17   **Click Finished.**

18   **If you elected to use the Console, the Sun Java System Console Login dialog box is displayed (see).**
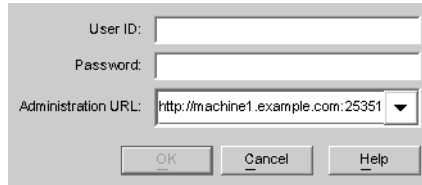
FIGURE 5–9 Logging into the Console

You must enter the following information to log into the Console:

- **User ID**: Enter the Administrator's user ID you specified when you installed the Administration Server on your machine.

- **Password**: Enter the Administrator's password specified during Administration Server installation.

- **Administration URL**: Enter the Administration Server's current URL location using the following format:

  http://*hostname.your_domain.domain:port_number*

  Where:

  - *hostname.your_domain.domain* is the computer host name you selected when you installed Administration Server.

  - *port_number* is the port you specified for Administration Server.

19 **After providing your credentials, click OK to close the dialog box.**

20 **You will then be prompted for the configuration password. Enter the password and click OK.**

When the Sun Java System Server Console window is displayed, you can start configuring Core. Continue to Chapter 6 for instructions.

6

Configuring Core Resources

You must initially configure the Core resources immediately after installing the Identity Synchronization for Windows Core.

This chapter explains how to add and configure these resources using the Console, and is organized into the following sections:

Note – To effectively configure Core resources you must know how to configure and operate Directory Server and Active Directory.

You are not required to configure these resources in a particular order (unless specifically noted in the text); however, using the configuration order presented in this chapter until you become more familiar with the product can save time and prevent errors.

Configuration Overview

This section illustrates the steps you will use to configure the Core resources for your deployment.
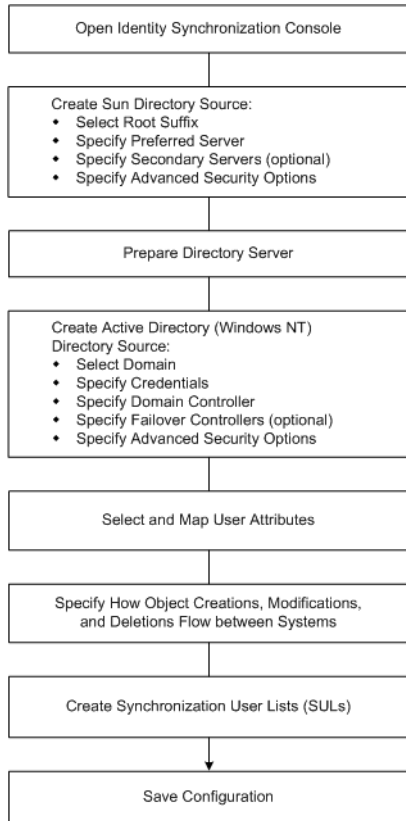
FIGURE 6–1   Configuring Core Resources for Your Deployment

Opening the Identity Synchronization for Windows Console

The Sun Java System Server Console window lists all of the servers and resources under your control and provides information about your system.
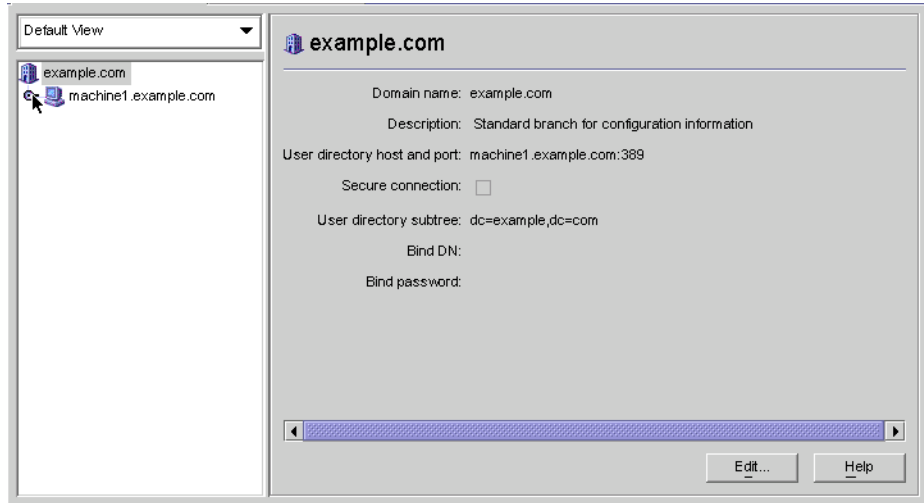
FIGURE 6–2   Sun Java System Server Console

Note – If you have not logged into the Sun Java System Server Console yet, return to Figure 5–9 for instructions.

▼ To Open Identity Synchronization for Windows Console

1   **On the Servers and Applications tab, select the hostname node in the navigation tree that contains the Server Group to which the Identity Synchronization for Windows instance belongs.**

2   **Expand the Server Group node and select the Identity Synchronization for Windows node.**
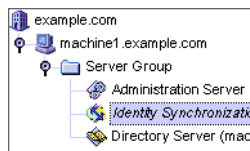
FIGURE 6–3   Expanding the Server Group

The information panel changes to provide information about Identity Synchronization for Windows and your system.
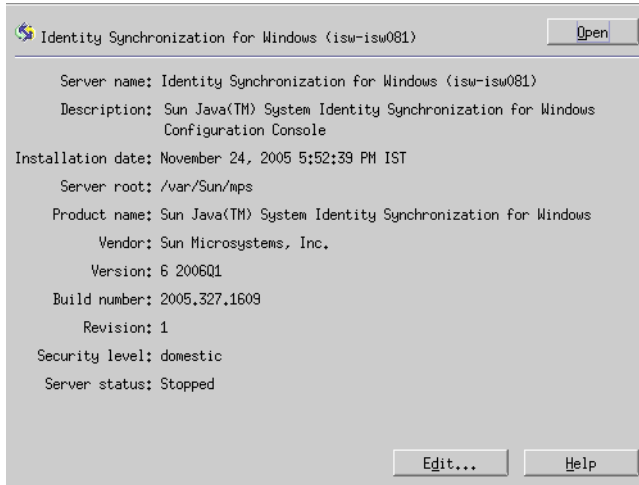
FIGURE 6–4   Information Panel

3   Click the Open button (located in the upper-right corner of the panel).

Note – The Edit button (located at the bottom of the panel) enables you to edit the Server name and Description.

---

4   You will be prompted to enter the configuration password that you specified during Core installation. Enter the password and click OK.

The Identity Synchronization for Windows Console is displayed, as follows:
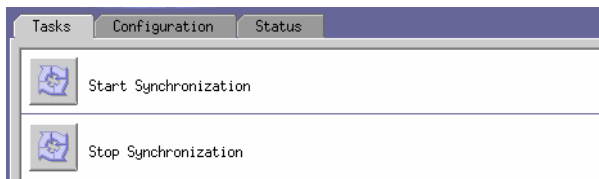
FIGURE 6–5   Console: Tasks Tab

This window contains three tabs:

- **Tasks** (*Default*): Use this tab to stop and start synchronization between your Sun and Windows systems. (Information about starting and stopping services is provided in "Starting and Stopping Synchronization" on page 218)

Note – Do not confuse starting and stopping Synchronization Services with starting and stopping Windows services.

To start or stop Windows services, you must do so from the Windows Console by selecting Start → Console → Administrative Tools → Computer Management → Services.

- **Configuration**: Use this tab to configure your systems for synchronization.
- **Status**: Use this tab to do the following:
  - Monitor the status of system components (such as Connectors).
  - View the audit and error logs generated by Identity Synchronization for Windows during configuration and synchronization.
  - Update and check the installation and configuration To Do list.

5   Select the Configuration tab.
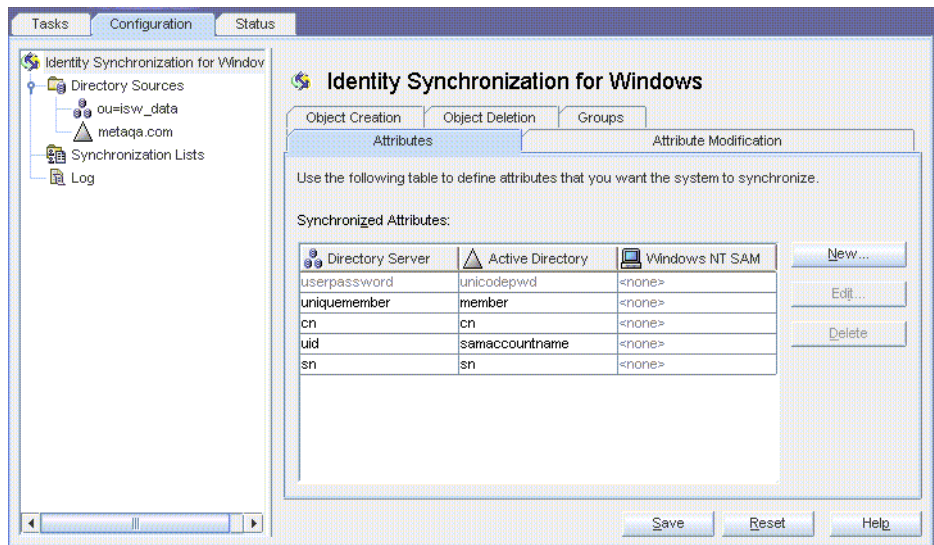
FIGURE 6–6   Console: Configuration Tab

The Configuration panel consists of the following tabs:

- **Attributes**: Use this tab to specify the attributes you want to synchronize between systems.
  - **Attribute Modification**: Use this tab to specify how passwords, attribute modifications, and object disablements are propagated between systems.
  - **Object Creation**: Use this tab to specify how newly created passwords and attributes are propagated between systems, and to specify initial values for the objects created by Identity Synchronization for Windows during synchronization.

- **Object Deletion**: Use this tab to specify how deleted passwords and attributes are propagated between systems.

  You must configure at least one Sun Java System Directory Server directory source, and at least one Windows server directory source (either Active Directory or Windows NT). Proceed to the next section for instructions.

Creating Directory Sources

▼ To Create Directory Sources

You must create directory sources in the following order (based on which sources you will be synchronizing).

1 **"Creating a Sun Java System Directory Source" on page 146**

2 **"Preparing Sun Directory Source" on page 153**

3 **"Creating an Active Directory Source" on page 156**

4 **"Creating a Windows NT SAM Directory Source" on page 163**

Note – At minimum, you must configure at least one Sun Java System Directory source and at least one Windows directory source (Active Directory and/or NT SAM).

Select the Directory Sources node in the navigation tree and the Directory Sources panel is displayed.
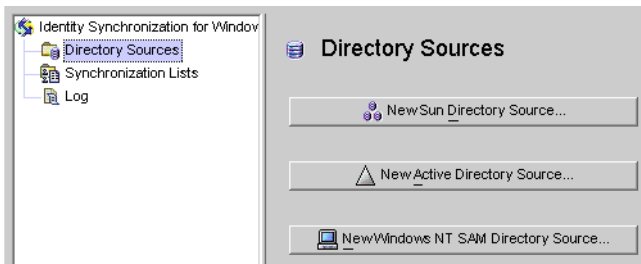
FIGURE 6–7   Accessing the Directory Sources Panel

Creating a Sun Java System Directory Source

Each Sun Java System directory source is associated with a Connector and set of Plug-ins that can be deployed in a replication scenario involving multiple servers. The Directory Server

Connector is capable of synchronizing changes from Windows directory source to the preferred server (master). In case, the preferred server is down, the changes will failover to the secondary server in the configured secondary servers list in a sequential manner till the preferred server comes up. Directory Server replication will replicate changes made from the preferred server (master) to other preferred secondary servers configured in the topology. Any Directory Server Plug-in can handle password validity checks from Windows directory sources and users can change passwords at any server.

▼ To Create a New Sun Java System Directory Source

1 Click the New Sun Directory Source button to invoke the Define Sun Java System Directory Source wizard.
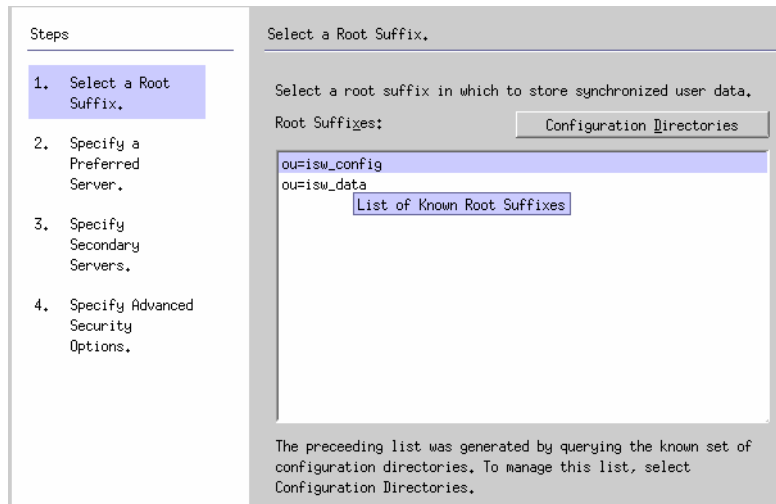
FIGURE 6–8 Selecting a Root Suffix

The program queries a known set of configuration directory sources and displays existing root suffix (also referred to as *naming contexts*) in the list pane.

By default, the program knows about the configuration directory where you installed the product, and the root suffixes known by the configuration directory will be listed in the list pane.

2 Select the root suffix where your users are located from the list pane. (If several root suffixes are listed, select the one where your users are located.) Click Next.

If the root suffix you want to synchronize with is not affiliated with a configuration directory registered with Identity Synchronization for Windows, then you must specify a new configuration directory, as follows:

a. Click the Configuration Directories button to specify a new configuration directory.

b. When the Configuration Directories dialog box is displayed (Step 3), click the New button to open the New Configuration Directories dialog box.
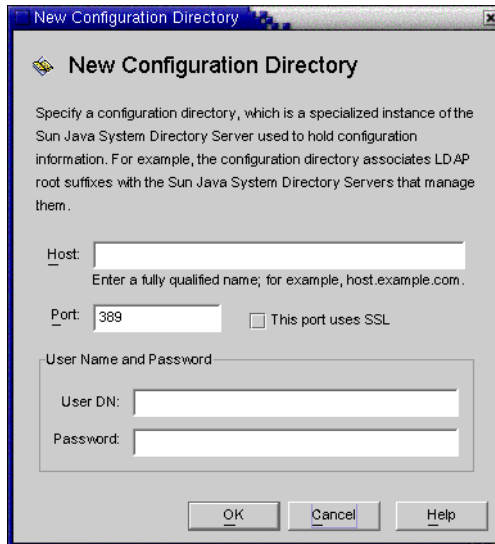
FIGURE 6–9 Selecting a New Configuration Directory

c. Enter the following information, and then click OK to save your changes and close the dialog box.

- **Host**: Enter the fully qualified host name.

    For example: `machine1.example.com`

- **Port**: Enter a valid, unused LDAP port number. (*Default is 389.*)

    Enable the This port uses SSL box if Identity Synchronization for Windows is using an SSL (Secure Socket Layer) port to communicate with the configuration directory.

- **User DN**: Enter your Administrator's (bind) distinguished name. For example,
    `uid=admin,ou=Administrators,ou=TopologyManagement,o=NetscapeRoot`

- **Password**: Enter your Administrator's password.

The wizard will query the specified configuration directory to determine all of the directory servers managed by that directory.

Note – Identity Synchronization for Windows only supports one root suffix per Sun Java System Directory Server source.

Editing and Removing Configuration Directories

You can also use the Configuration Directories dialog box to manage your list of configuration directories, as follows:

- Select a configuration directory from the list pane, and then click the Edit button. When the Edit Configuration Directories dialog is displayed, you can change the Host, Port, Secure Port, User Name, and Password parameters.

- Select a configuration directory from the list pane, and then click Remove to delete the directory from the list.

d. Click OK to close the Configuration Directories dialog box and the newly selected configuration directory's root suffixes are displayed in the list pane.

By default, Directory Server creates a root suffix whose prefix corresponds to the components of the machine's DNS domain entry. It uses the following suffix:

dc=*your_machine's_DNS_domain_name*

That is, if your machine domain is *example.com*, then you should configure the suffix dc=example, dc=com for your server. The entry named by the chosen suffix must already exist in the directory.

e. Select the root suffix, and click Next.

The Specify Preferred Servers panel is displayed (see "Creating a Sun Java System Directory Source" on page 146).
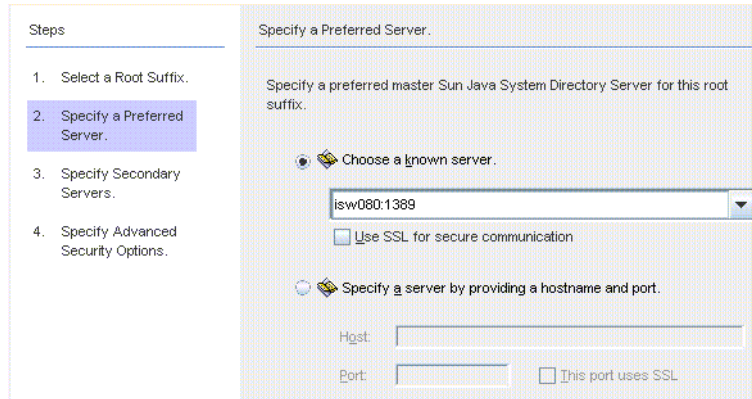
FIGURE 6–10 Specifying a Preferred Server

Identity Synchronization for Windows uses the preferred Directory Server to detect changes made at any Directory Server master. The preferred server also acts as the primary location where changes made on Windows systems are applied to the Sun Java System Directory Server system.

If the preferred master server fails, the secondary server can store these changes until the preferred server (master) comes back online.

3 Use one of the following methods to select a preferred server:

- Select the Choose a Known Server option, and then select a server name from the drop-down list.

Note – The Directory Server must be running to appear in the list. If the server is down temporarily, select the Specify a Server by Providing a Hostname and Port option, and then enter the server information manually.

Enable the Use SSL for secure communication box if you want the Directory Server to communicate using SSL. However, if you enable this feature there are some additional setup steps you must perform after installation. For more information, see "Enabling SSL in Directory Server" on page 241

- Select the Specify a Server By Providing a Hostname and Port option, and then type the server's Host name and Port into the text fields.

Select the This Port Uses SSL checkbox if the port you specified uses SSL.

4 Click Next and the Specify a Secondary Server panel is displayed.
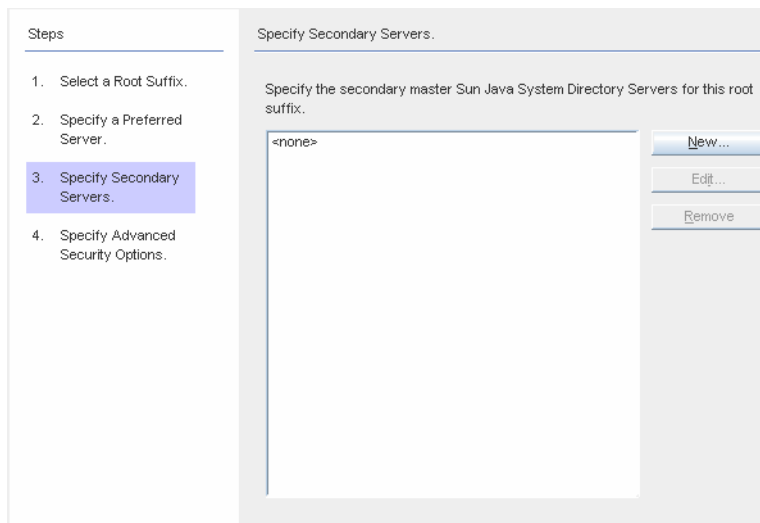
FIGURE 6–11 Specifying the Secondary Servers for Failover Support

You can add, edit, or delete the Secondary Servers:

- **Click the New button to display the Add Sun Directory Source dialog box. Enter the host name, port, user DN, password, and then click OK. For more information on these fields, see Step c.**

- **Click the Edit button to display the Edit Sun Directory Source dialog box. Enter the host name, port, user DN, password, and then click OK. For more information on these fields, see Step c.**

- **From the Secondary Servers list, select the server you want to delete and click the Remove button.**

5 **To specify the secondary Directory Servers, select a server name from list, and then click Next.**

Note –

- The Directory Server must be running or the server name will *not* appear in list.

- Do not use the same host name and port for both the preferred and the secondary servers in a Sun directory source.

- If you enable the Secure Port feature, there are additional setup steps you must perform after installation. For more information, see "Enabling SSL in Directory Server" on page 241

If you do not want to specify a secondary server, click Next.

---

6 If you want to use secure SSL communication, *read the notes below*, **and then enable one or both of the following options:**
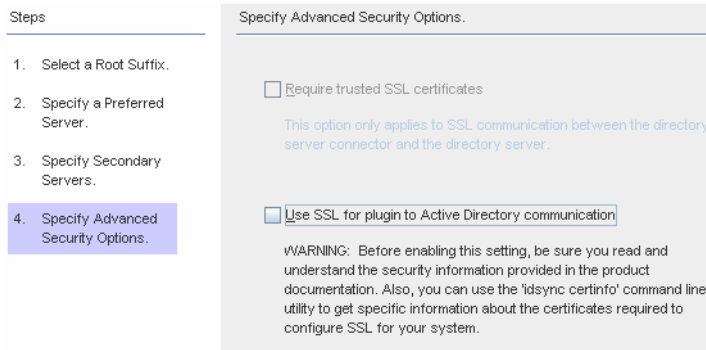
FIGURE 6–12   Specifying Advanced Security Options

Note – You must install the Directory Server Plug-in on each Directory Server (any master, replica, or hub) where users will bind or where passwords will be changed.

When the Directory Server Plug-in synchronizes passwords and attributes to Active Directory, it must bind to Active Directory to search for users and their passwords. In addition, the Plug-in writes log messages to the central log and into the Directory Server's log. By default these communications are not accomplished over SSL.

- To encrypt channel communication only or to encrypt channel communication and use certificates to ensure participants' identity verification between Directory Server and the Directory Server Connector, enable the Require Certificates for SSL box.

  Clear the checkbox if you do not want to trust certificates.

- To use secure SSL communication between the Directory Server Plug-in and Active Directory, enable the Use SSL for Plug-in to Active Directory communication box.

If you enable these features, then additional setup is required after installation. See "Enabling SSL in Directory Server" on page 241

- You can use the use the idsync certinfo command line utility to determine which certificates you must add for each Directory Server Plug-in and/or Connector certificate database. See "Using certinfo" on page 290

- If your primary and secondary Directory Servers are part of a multimaster replication (MMR) deployment, refer to Appendix E

7 When you are finished with the Specify Advanced Security Options panel, click Finish.

The program adds the selected directory sources to the navigation tree under Directory Sources, and the Prepare Directory Server Now? dialog is displayed.

You must prepare the Directory Server to be used by Identity Synchronization for Windows. You can choose to perform this task now, or you can do it later — but you must prepare the Directory Server before you install the Connectors. (Instructions for installing Connectors are provided in Chapter 7).

- If you want to prepare the Directory Server now, click Yes to open the wizard, and then proceed to the next section, "Preparing Sun Directory Source" on page 153

- If you prefer to perform this process later, click No and proceed to "Creating an Active Directory Source" on page 156.

Preparing Sun Directory Source

This section explains how to prepare Sun Directory source for use by Identity Synchronization for Windows.

Preparing the Directory Server:

- Creates the Retro-Changelog database and access control instance available on the preferred host

- Creates the Connector user and user access control instance available on the preferred host

- Creates an equality index on the preferred and secondary hosts

---

Note –

- As an alternative to using the Console, you can use the `idsync prepds` command line utility to prepare the Directory Server. For more information, see "Using prepds" on page 292.

- To prepare the Directory Server using the `idsync prepds` command line utility, you must know which hosts and suffixes you will be using and you must have Directory Manager's credentials.

  You can use the Prepare Directory Server wizard to prepare the Directory Server.
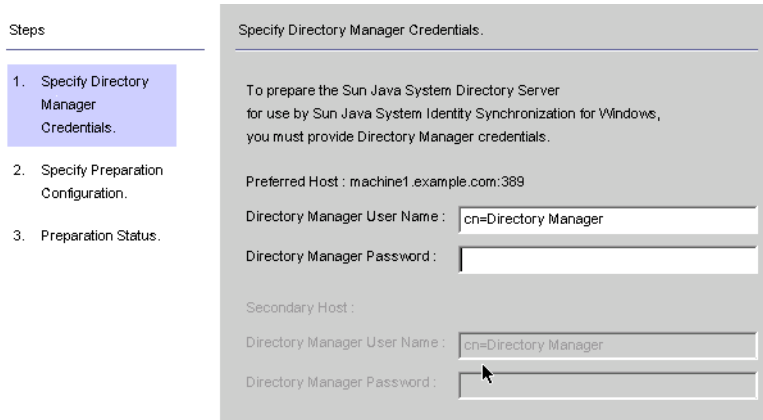
FIGURE 6–13 Entering Your Directory Manager Credentials

Note – To access this wizard, use one of the following methods:

- When the Prepare Directory Server Now? dialog box is displayed, click the Yes button.
- When the Sun Directory Sources panel is displayed (on the Configuration tab), click the Prepare Directory Server button.

▼ To Prepare your Directory Server Source

1 Enter the following credentials for the Directory Manager account.

- **Directory Manager User Name**
- **Directory Manager Password**

  If you are using a secondary host (MMR configurations), then the Secondary Host options will be active and you must specify credentials for these hosts too.

2 When you are done, click Next and the Specify Preparation Configuration panel is displayed.
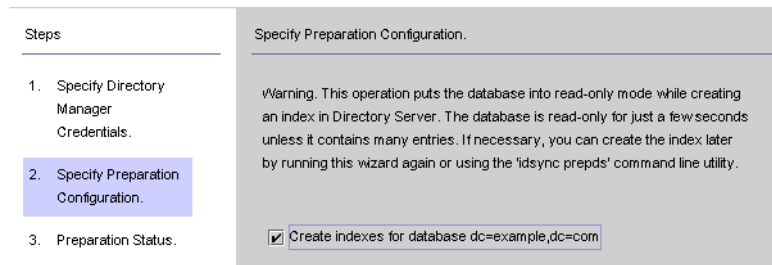
FIGURE 6–14 Specifying the Preparation Configuration

Read the warning, and then decide whether to create the Directory Server indexes now or later.

Note –

- This operation can take some time, depending on the size of your database.
- While your database is in read-only mode, any attempts to update information in the database will fail.
- Taking your database off-line enables you to create the indexes much faster.
- To create the indexes now, enable the Create indexes for database box, and then click Next.
- To create the indexes later (either manually or by running this wizard again) clear the Create indexes for database box, and then click Next.

---

3. **The Preparation Status panel is displayed to provide information about the Directory Server preparation progress.**

   - When a SUCCESS message is displayed at the bottom of the message pane, click Finish.
   - If error messages display, you must correct the problem(s) reported before you can continue. Check the error logs (see the Status tab) for more information.

4. **Return to the Configuration tab in the Console. Select the Sun Directory source node in the navigation tree to view the Sun Directory Source panel.**
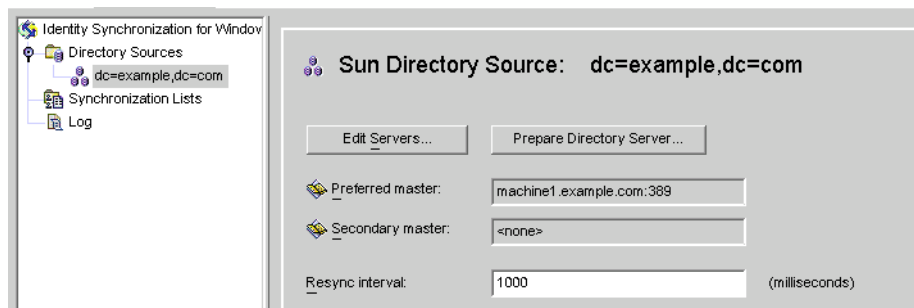
FIGURE 6–15   Sun Directory Source Panel

From this panel, you can perform the following tasks:

- **Edit servers**: Click this button to reopen the Define Sun Java System Directory Source panel where you can change any of the server configuration parameters. If necessary, review the instructions provided for "Creating a Sun Java System Directory Source" on page 146.

> **Note –** If you re-create the Retro-Changelog database for the preferred Sun directory source, the default access control settings will not allow the Directory Server Connector to read the database contents.
>
> To restore the access control settings for new the Retro-Changelog database, run `idsync prepds` or click the Prepare Directory Server button after selecting the appropriate Sun directory source in the Console.

- **Prepare Directory Server**: Click this button and follow the instructions for "Preparing Sun Directory Source" on page 153 to prepare a Directory Server.

  If anything changes on the Directory Server after you initially prepare the server (for example, if an index is deleted or you lose the Retro-Changelog database), you can re-prepare the server.

- *Resync interval*: Specify how often you want the Directory Server Connector to check for changes. (*Default is 1000 milliseconds*.)

5   Add a Directory Server directory source for each user population in your Sun Java System Directory Server enterprise that you want to synchronize.

When you are finished, you must create at least one Windows directory source:

- To create an Active Directory directory source, continue to the next section, "Creating an Active Directory Source" on page 156.
- To create a Windows NT directory source, continue to "Creating a Windows NT SAM Directory Source" on page 163

Creating an Active Directory Source

You should add an Active Directory directory source for each Windows domain in your network that you want to synchronize.

Each Active Directory deployment has at least one global catalog that knows about all the global information across all Active Directory domains. To access the global catalog, the rights assigned to a normal user are sufficient unless you change the default permissions.

> **Note –** It is possible for each Active Directory server to be a global catalog and a deployment can have multiple global catalogs, but you only need to specify one global catalog.

▼ To Configure and Create Windows Active Directory Servers in a Network

Perform these steps if there are Windows Active Directory servers in your network:

1 **Select the Directory Sources node in the navigation tree, and then click the New Active Directory Source button on the Directory Sources panel.**

The Windows Global Catalog dialog box is displayed.

FIGURE 6–16 Windows Global Catalog

2 **Enter the following information and then click OK:**

- **Host**: Enter the fully qualified host name of the machine that holds the global catalog for the Active Directory forest.

  For example: **machine2.example.com**

- **This port uses SSL**: Enable this option if Identity Synchronization for Windows is using an SSL port to communicate with the global catalog.

- **User DN**: Enter your fully qualified Administrator's (bind) distinguished name. (Any credentials that enable you to browse the schemas and determine which Active Directory domains are available on your system will suffice.)

  For example: **cn=Administrator,cn=Users,dc=example,dc=com**

- **Password**: Enter a password for the specified user.

3 **The Define Active Directory Source wizard is displayed, as follows.**
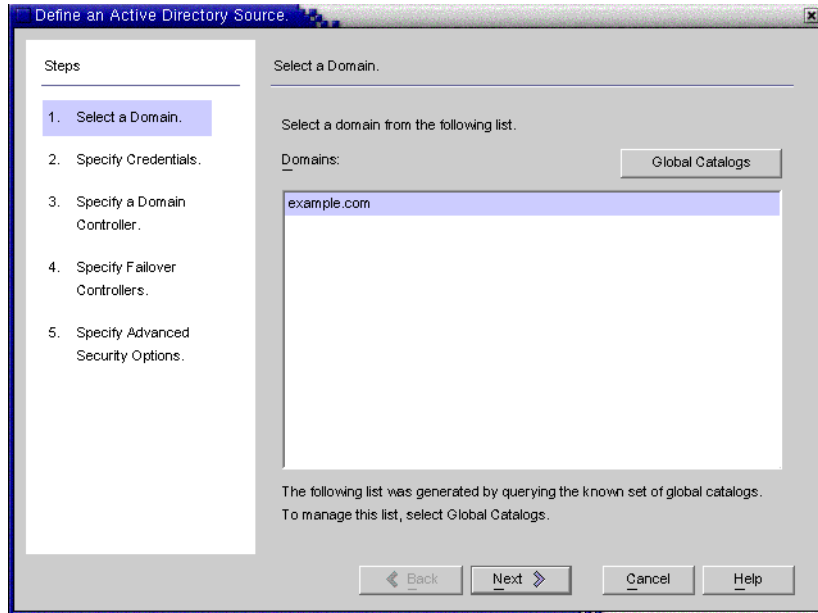
FIGURE 6–17 Define an Active Directory Source Wizard

This wizard queries the Active Directory global catalog to determine what other domains exist, and displays those domains in the Domains list pane.

4 Select a name from the list pane to specify an Active Directory domain and click OK.

If the domain you want to use is not displayed in the list, you must add the global catalog that knows about that domain using the following steps:

a. Click the Global Catalogs button and the Global Catalogs wizard is displayed.
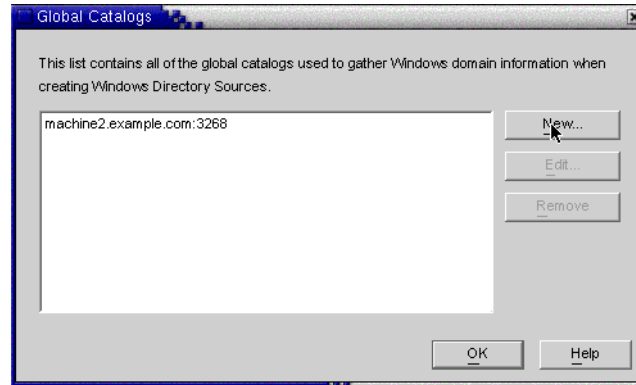
FIGURE 6–18   Specifying a New Global Catalog

    b. Click the New button.

    c. When the Windows Global Catalog dialog box is displayed, provide the global catalog's Host name and your Directory Source credentials (as described in Step 2), and then click OK

    d. The new global catalog and port, are displayed in the Global Catalogs list panel. Select the catalog name, and then click OK.

    e. Repeat these steps if you want to add more global catalogs (domains) to the system.

    f. When you are done, click the Next button in the Select a Domain pane.

5   When the Specify Credentials panel is displayed, review the value in the User DN field.
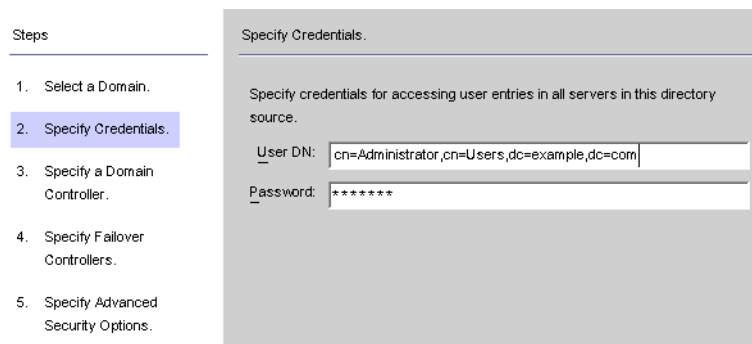
FIGURE 6–19   Specifying Credentials for This Active Directory Source

If the program did not automatically enter the Administrator's distinguished name in the User DN field (or you do not want to use the Administrator's credentials) enter a User DN and password manually.

When configuring an Active Directory source, you must provide a user name and password that the Active Directory Connector can use to connect to Active Directory.

Note – The Connector requires specific access rights. Minimum rights will depend on the direction of synchronization, as follows:

- If you are configuring synchronization flow from Active Directory to Directory Server only, then the user provided for the Active Directory Connector does not require many special privileges. A normal user with the extra privilege to "Read All Properties" in the domain being synchronized will suffice.

- If you are configuring synchronization flow from Directory Server to Active Directory, then the Connector user must have more privileges because, synchronization changes the user entries in Active Directory. In this setup, the Connector user must have either the "Full Control" privilege or be a member of the Administrators group.

6   Click Next to open the Specify a Domain Controller panel.
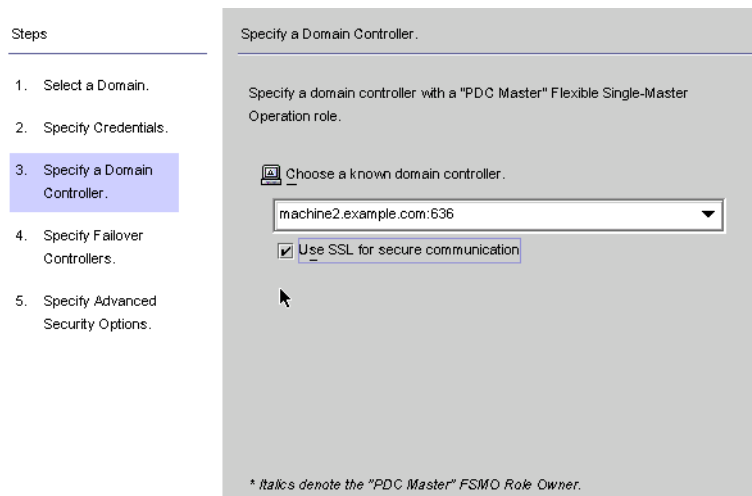
FIGURE 6–20   Specifying a Domain Controller

Use this panel to select a controller to synchronize within the specified domain. (The domain controller is similar in concept to a Directory Server's preferred server.)

If the selected Active Directory domain has multiple domain controllers, select the domain controller with the Primary Domain Controller flexible single master operation (FSMO) role for synchronization.

By default, password changes made at all domain controllers will be replicated immediately to the Primary Domain Controller FSMO role owner, and if you select this domain controller, Identity Synchronization for Windows will synchronize these password changes immediately to the Directory Server.

In some deployments, the AvoidPdcOnWan attribute may be set in the Windows registry because there is a significant network "distance" to the PDC, which will delay synchronization significantly. (See *Microsoft Knowledge Base Article 232690* for more information.)

7 Select a domain controller from the drop-down list.

8 If you want the Identity Synchronization for Windows Connector to communicate with the domain controller over a secure port, enable the Use a Secure Port box.

Note – The program automatically installs the CA certificate in the Active Directory Connector if you are using Microsoft certificate server. If you are not, then you must manually add the CA certificate in the Active Directory Connector (see "Enabling SSL in the Active Directory Connector" on page 243 change your flow settings after initial configuration these procedures apply as well.

---

9 When you are done, click Next.

The Specify Failover Controllers panel is displayed (see "Creating an Active Directory Source" on page 156). You can use this panel to specify any number of failover domain controllers.
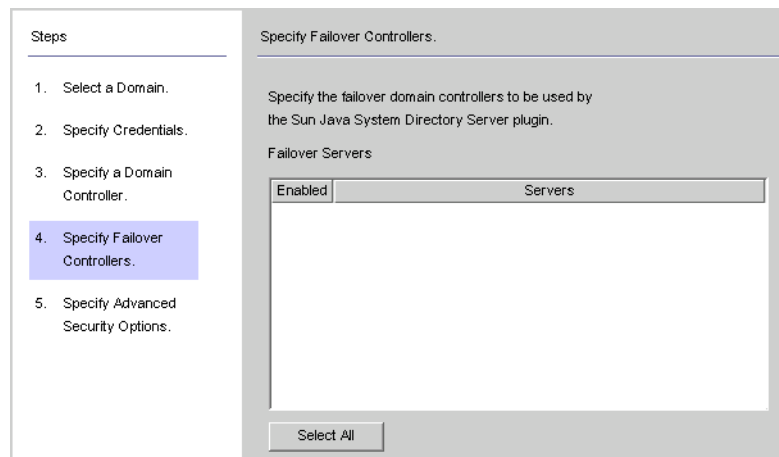
FIGURE 6–21 Specifying Failover Controllers

The Active Directory Connector communicates with only one Active Directory domain controller, and Identity Synchronization for Windows does not support failover changes applied by that Connector. However, the Directory Server Plug-in will communicate with any number of domain controllers when validating password changes to Directory Server.

If Directory Server tries connecting to an Active Directory domain controller and that domain controller is not available, Directory Server will iteratively try connecting to the failover domain controller(s) specified.

10 **Select one or more of the server names listed in the Failover Servers list pane (or click the Select All button to specify all of the servers in the list), and then click Next.**

11 **The Specify Advanced Security Options panel is displayed.**

The Require trusted SSL certificates option is active (available for selection) only if you enabled the Use SSL for Secure Communication box on the Specify a Domain Controller panel.
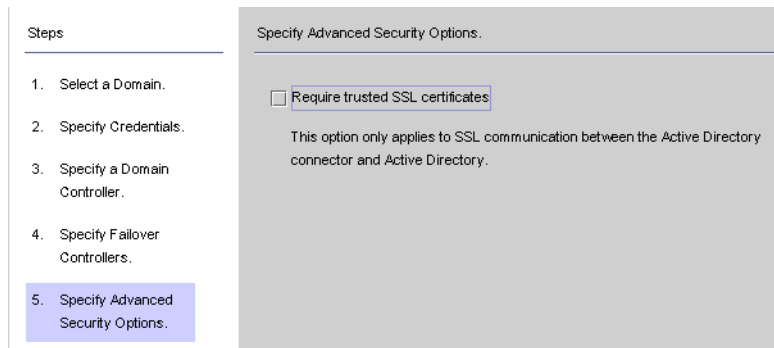
FIGURE 6–22 Specifying Advanced Security Options

- If the Require trusted SSL certificate box is disabled (*Default setting*), the Active Directory Connector will connect to Active Directory over SSL and does not verify that it trusts the certificates passed by Active Directory.

  Disabling this option simplifies the set-up process because you do not have to put an Active Directory Certificate in the Active Directory certificate database.

- If you enable the Require trusted SSL certificate box, the Active Directory Connector will connect to Active Directory over SSL and it must verify that it trusts the certificates passed by Active Directory.

Note – You must add Active Directory Certificates to the Active Directory Connector's certificate database. For instructions, see "Adding Active Directory Certificates to the Connector's Certificate Database" on page 245.

---

12 **When you are finished with the Advanced Security Options panel, click the Finish button.**

The program adds the newly specified Active Directory source to the navigation tree under Directory Sources.

13 **Select the Active Directory directory source node to view the Active Directory Source panel.**
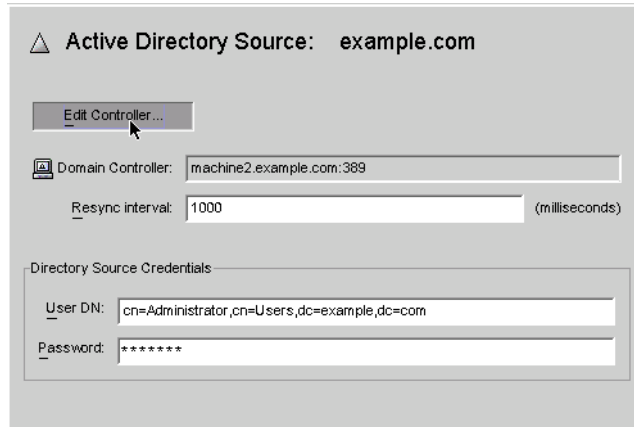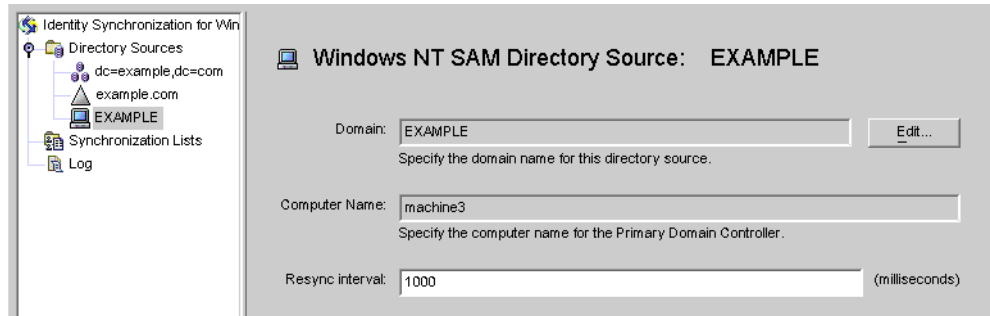
FIGURE 6–23   Active Directory Source Panel

From this panel, you can perform the following tasks:

- **Edit Controllers**: Click this button to reopen the Specify a Domain Controller panel where you can change any of the domain controller configuration parameters. If necessary, review the instructions provided for "Creating an Active Directory Source" on page 156.
- **Resync Interval**: Specify how often you want the Active Directory Connector to check for changes. (*Default is 1000 milliseconds.*)
- **Directory Source Credentials**: Change the specified User DN and/or password.

Creating a Windows NT SAM Directory Source

This section explains how to create a Windows NT SAM Directory Source where you can deploy Identity Synchronization for Windows.

▼ To Deploy Identity Synchronization for Windows on Windows NT

1   Select the Directory Sources node in the navigation tree, and then click the New Windows NT SAM Directory Source button.
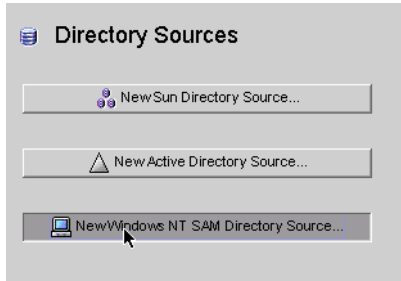
FIGURE 6–24   Directory Sources Panel

2   When the Define a Windows NT SAM Directory Source panel is displayed, follow the instructions for locating the Windows NT domain name, and enter the unique NT directory source domain name in the Domain field. When you are done, click Next.
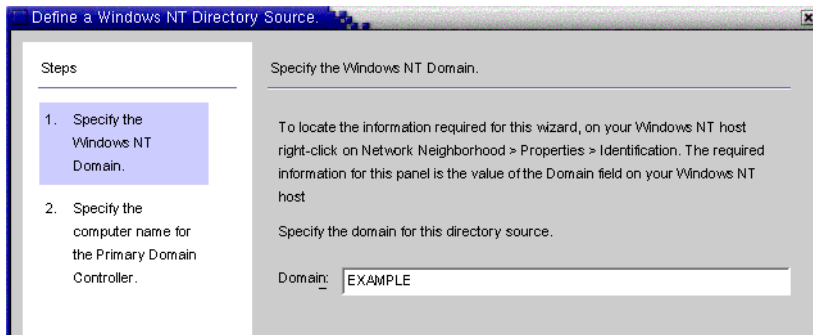
FIGURE 6–25   Specifying a Windows NT SAM Domain Name

3   When the Specify the Computer Name for the Primary Domain Controller panel is displayed, follow the instructions for locating the Primary Domain Controller computer name, and enter the information in the Computer Name field.
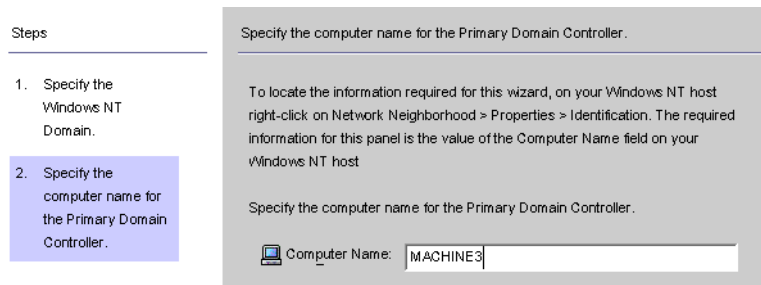
FIGURE 6–26   Specifying a Name for the Primary Domain Controller

4   Click Finish.

The program adds the newly specified Windows NT SAM directory source to the navigation tree under Directory Sources. Select the new directory source node to view the Windows NT SAM Source panel.

FIGURE 6–27   Windows NT SAM Directory Source Panel

From this panel, you can perform the following tasks:

- *Edit*: Click this button to reopen the Specify a Domain Controller panel where you can change any of the domain controller configuration parameters. If necessary, review the instructions provided for "Creating an Active Directory Source" on page 156.

- *Resync interval*: Specify how often you want Identity Synchronization for Windows to check for changes made on Windows NT. (*Default is 1000 milliseconds.*)

5   Add a Windows NT directory source for each Windows NT machine in your network.

When you are finished creating Windows NT SAM directory sources, you are ready to create and map attributes to be synchronized, continue to "Selecting and Mapping User Attributes" on page 165

Selecting and Mapping User Attributes

After you have created and configured your Directory Server and Windows directory sources, you must decide which user attributes you want to synchronize and then map those attributes between systems.

The information in this section is organized as follows:

Selecting and Mapping Attributes

There are two types of attributes:

- **Significant**: Attributes that are synchronized between systems when you create or modify user entries.

- **Creation**: Attributes that are synchronized between systems only when you create user entries.

  Some creation attributes are *mandatory* based on the schema used for each platform. These attributes are required for password synchronization and they must be mapped to Directory Server attributes to successfully create a user object class entry on the Active Directory server.

  This section explains how to select user attributes for synchronization and how to map these attributes (one-to-one) so that when you specify an attribute for Directory Server the equivalent attribute will display in your Active Directory and/or Windows NT environment (and vice versa), and the companion Windows attributes will have their values synchronized.

▼ To Select and Map Attributes for Synchronization

1 Select the Identity Synchronization for Windows node at the top of the navigation tree.
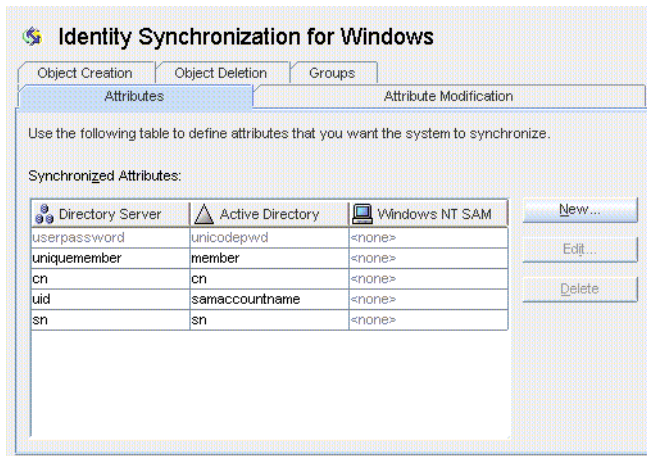
FIGURE 6–28   Attributes Tab

Note – When the Group Synchronization feature has been enabled, the *uniquemember* (Directory Server) attribute and *member attribute* (Active Directory) are internally mapped and would be indicated as shown in the console.

2 **Select the Attributes tab and then click the New button.**

The Define Significant Attribute Mappings dialog box is displayed. Use this dialog box to map attributes from Directory Server to your Windows Systems (Active Directory and/or Windows NT).
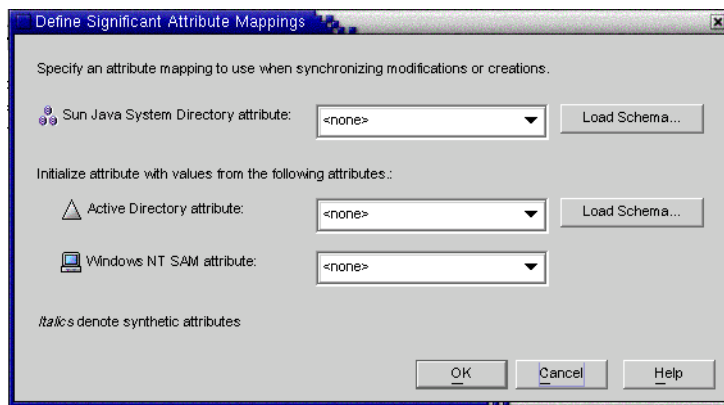
FIGURE 6–29 Defining Significant Attribute Mappings

Note – Which creation attributes are mandatory for Directory Server (or for Active Directory) will depend on the objectclass configured for your Sun-side (or Active Directory-side) user entries.

The program automatically uses *inetOrgPerson* as the default objectclass for Directory Server, and you loaded the Active Directory schema when you specified the global catalog. So you do not use the Load Schema buttons unless you want to change the default schema.

If you want to change the default schema source, see "Changing the Schema Source" on page 169

3 **Select an attribute from the Sun Java System attribute drop-down list (for example***cn***), and then select the equivalent attribute from the Active Directory attribute and/or Windows NT SAM attribute drop-down menus.**

4 **When you are finished, click OK.**

5   **To designate additional attributes, repeat steps 2 through step 4.**

A finished Synchronized Attributes table might look something like the following example, which shows the userpassword, cn, and telephonenumber Directory Server attributes mapped to unicodepwd, cn, and telephonenumber Active Directory attributes.

Directory Server	Active Directory	Windows NT SAM
userpassword	unicodepwd	user_password
cn	cn	<none>
telephonenumber	telephonenumber	<none>

FIGURE 6–30   Completed Synchronized Attributes Table

Creating Parameterized Default Attribute Values

Identity Synchronization for Windows allows you to create *parameterized* default values for attributes using other creation or significant attributes.

To create a parameterized default attribute value, you embed an existing creation or significant attribute name— preceded and followed by percent symbols (% *attribute_name* %) — in an expression string. For example, homedir=/home/%uid% or cn=%givenName% %sn%.

When you create these attribute values:

- You can use multiple attributes in a creation expression (cn=%givenName% %sn%).
- If A=0, then B can have one default value only.
- You can use the backslash symbol (\\) for quoting (for example, diskUsage=0\\%).
- Do not use expressions that have cyclic substitution conditions (for example, if you specify description=%uid%, you cannot use uid=%description%..)

Note – When Group Synchronization is enabled, the following are important:

1. The creation expression supported at Active Directory is cn=%cn%.

2. The creation expression must contain valid attribute names belonging to the group objectclass also since the creation expression is common to both user as well as the group.

   For example: The attribute sn is not part of the groupofuniquenames objectclass at the Directory Server. Hence the following creation expression would be invalid for a group object. (Though it would work fine for user.)

   cn=%cn%.%sn%

3. The attribute used in the creation expression must be provided with a value for every user/group entry created. The value maybe provided using the command line interface, if the console does not have the provision.

Changing the Schema Source

The program automatically provides default schema sources, but allows you to change the default schema.

▼ To Change the Default Schema Source

1   Click the Load Schema button on the Define Significant Attribute Mappings dialog box.

The Select Schema Sources panel is displayed.
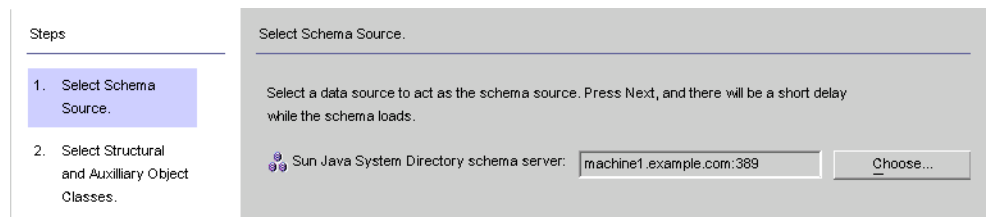
FIGURE 6–31   Selecting Schema Sources

Use this panel to specify from which Sun Java System Directory Server schema server you want to read the schema. This schema contains the object classes that are available on your system, and object classes define which attributes are available for users on your system.

The program adds your configuration directory to the Sun Java System Directory schema server field by default.

2   To select a different server, click the Choose button.

The Select a Sun Schema Host dialog box is displayed. This dialog box contains a list of the configuration directories that gather administrative information about your directory sources.

From this dialog box, you can:

- Create new configuration directories and add them to the list.

  Click New, and when the New Configuration Directory dialog box displays; specify a Host, Port, User DN, and Password. Click OK when you are done.

- Edit existing directories.

  Click Edit, and when the Edit Configuration Directory dialog box displays, you can change the Host, Port, User DN, and/or Password. Click OK when you are done.

- Remove directories from the list.

  Select a directory name from the list and then click the Remove button.

3   Select a server from the list and click OK when you are done. (Generally, one of your Sun synchronization host(s) is a good choice as a schema source.)

4   Click the Next button and the Select Structural and Auxiliary Object Classes panel is displayed.
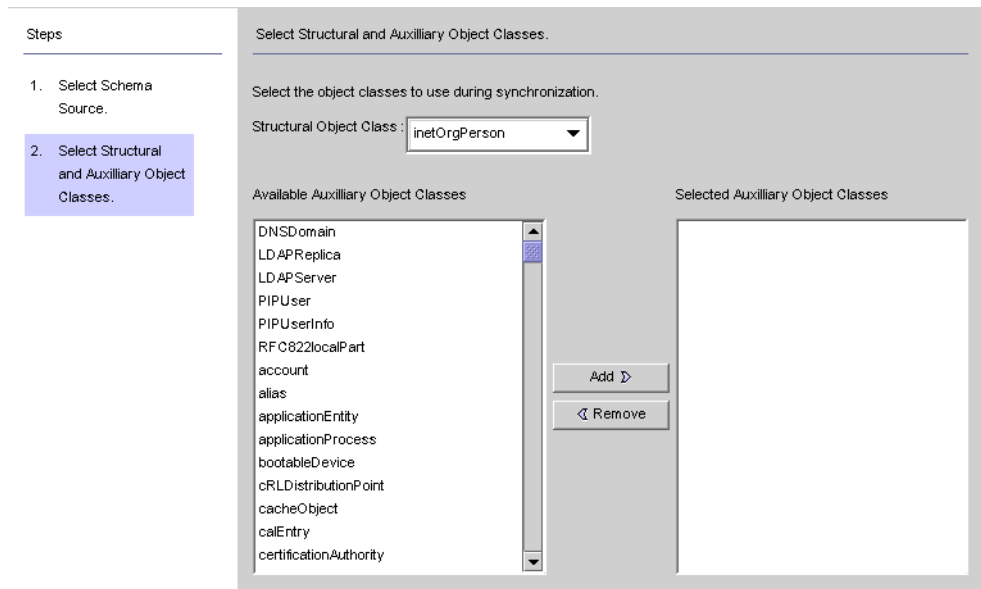
FIGURE 6–32   Selecting Structural and Auxiliary Object Classes

Use this panel to specify the object classes to synchronize, as follows:

- **Structural Object Class**: Every entry that is created or synchronized from the selected Directory Server must have at least one structural object class.

- **Auxiliary Object Classes**: These object classes augment the selected structural class and provide additional attributes for synchronization.

   To specify structural and auxiliary object classes:

a.   Select a structural object class from the drop-down list. (*Default is inetorgperson.***)**

b.   Select one or more object classes from the Available Auxiliary Object Classes list pane, and then click Add to move your selection(s) to the Selected Auxiliary Object Classes list pane.

   The selected object class(es) determine which Directory Server source attributes will be available for selection as significant or creation attributes. The object class(es) also determine the mandatory creation attributes.

   To delete selections from the Selected Auxiliary Object Classes list, click the object class name and then click the Remove button.

c.   When you are done, click Finish and the program loads the schema and selected object classes.

Propagating User Attributes Between Systems

After you create and map the user attributes you want to synchronize, you must tell Identity Synchronization for Windows how to propagate (flow) the attribute creations, modifications, and deletions between your Directory Server and Windows Systems.

By default, Identity Synchronization for Windows:

- Synchronizes from Windows to Directory Server only
- Synchronizes the password attribute only (unless you specified significant attributes in the previous section)
- Does not synchronize the creation or deletion of entries

    This section explains how to configure attribute synchronization between systems. The information is organized as follows:

Specifying How Object Creations Flow

▼ **To Specify How Object Creations Should Flow Between Directory Server and Active Directory Systems**

1   **Click the Object Creation tab.**
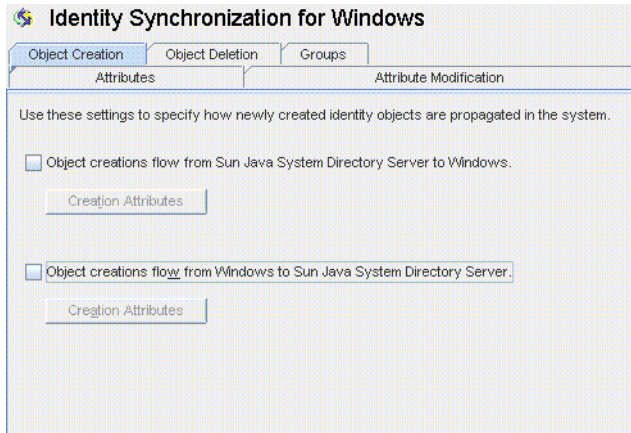
FIGURE 6–33   Selecting and Propagating Creations

2   You can enable or disable the flow of creations as follows:

- Enable **Object creations flow from Sun Java System Directory Server to Windows** to propagate creations from the Directory Server environment to your Windows servers.

- Enable **Object creations flow from Windows to Sun Java System Directory Server** to propagate creations from the Windows environment to your Directory Servers.

- Enable both options for bidirectional flow.

- Disable both options to prevent user creations from propagating from one system to the other. (*Default*).

3   To add, edit, or delete creation attributes to synchronize between systems, click the Creation Attributes button located under the selected option(s).

The Creation Attribute Mappings and Values dialog box displays.
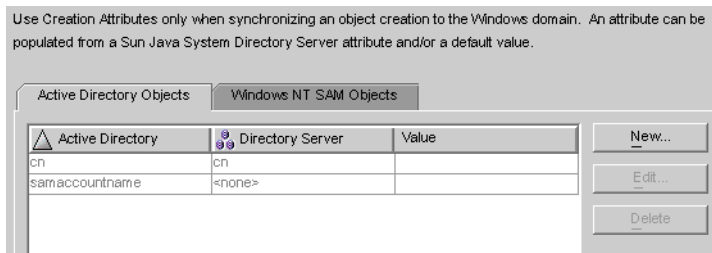
FIGURE 6–34   Creation Attributes Mappings and Values: Directory Server to Windows

Use Creation Attributes only when synchronizing an object creation to the Sun Java System Directory Server. An attribute can be populated from a Windows user attribute and/or a default value. You must specify a Creation Attribute for each mandatory attribute in the synchronized object class that is not already defined as a Synchronization Attribute.

Directory Server	Active Directory	Windows NT SA...	Value	
cn	cn	<none>		New...
sn	<none>	<none>		Edit...
				Delete

FIGURE 6–35   Creation Attributes Mappings and Values: Windows to Directory Server

You can use either of the dialog boxes to specify new creation attributes, edit, or delete existing attributes. For more information, see "Specifying New Creation Attributes" on page 173.

Note – To satisfy schema constraints regarding required attributes for user object classes, you may have to specify additional attributes to flow through the system during a user creation.

Additional attributes are not necessary if you specified the required attributes as *modification* attributes (as described in "Selecting and Mapping User Attributes" on page 165).

Specifying New Creation Attributes

The following instructions explain how to add and map creation attributes from Active Directory to Directory Server. (The procedure for adding and mapping creation attributes flowing from Directory Server to Windows and from Windows to Directory Server is similar.)

▼ To Specify New Creation Attributes

1   Click the New button in the Creation Attribute Mappings and Values dialog box.

The Define Creation Attribute Mappings and Values dialog box is displayed.
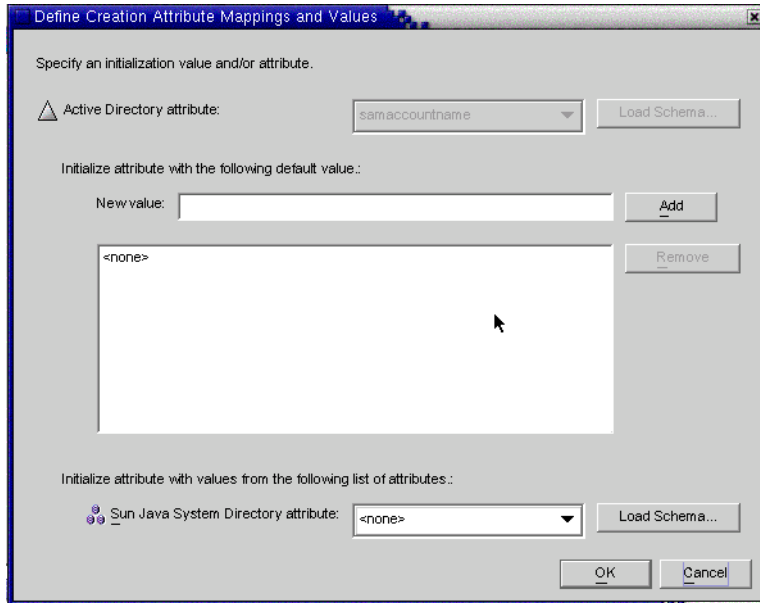
FIGURE 6–36 Defining Creation Attribute Mappings and Values

2 Select an attribute value from the Active Directory attribute drop-down list.

FIGURE 6–37 Selecting a New Active Directory Attribute

Identity Synchronization for Windows allows you to initialize an attribute with multiple values— if the attribute itself accepts multiple values.

For example, if your company has three fax telephone numbers, you can specify the facsilimiletelephonenumber attribute for both Sun Java System Directory Server and Active Directory, and specify the three numbers.

You must know which attributes will accept multiple values. If you try adding multiple values to an attribute that does not accept them, an error will result during runtime when the program attempts to create the object.

3 Enter a value in New value field and click Add.

The program adds the attribute value to the list pane. Repeat this step as many times as necessary to add multiple attribute values.
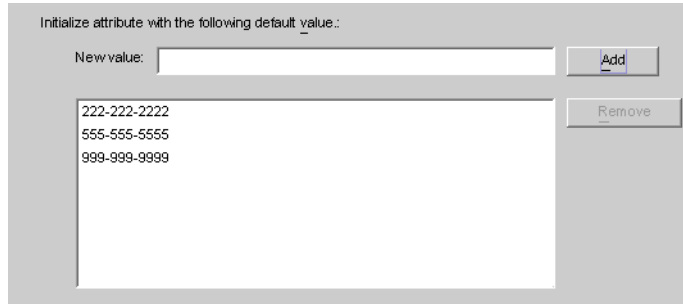
FIGURE 6–38 Specifying Multiple Values for a Creation Attribute

4 **To map the attribute to Directory Server, select an attribute name from the Directory Server attribute drop-down list.**
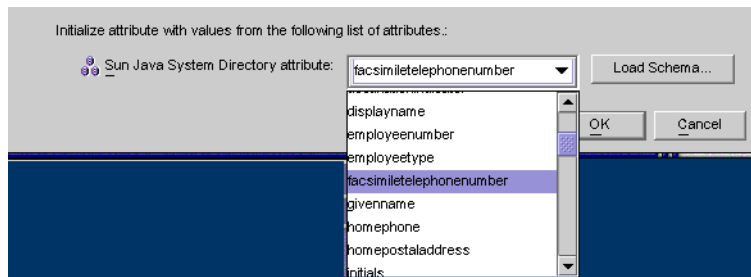
FIGURE 6–39 Mapping the Directory Server Attribute

5 **When you are finished, click OK.**

Based on the example, the finished Creation Attributes and Mappings table would look like the one in the following figure.

FIGURE 6–40 Completed Creation Attributes and Mappings Table

6 **To designate additional attributes, repeat these steps.**

Editing Existing Attributes

▼ To Edit Creation Attributes Mapping or Values

1 **Select the Object Creation tab, and click on the Creation Attributes button located under the selected creation option.**

2 **When the Creation Mappings and Values dialog box is displayed, select the attribute from the table, and then click the Edit button.**

The Define Creation Mappings and Values dialog box is displayed.

3 **Use the drop-down menus to change the existing mapping between Directory Server and Active Directory (or Windows NT).**

For example, if you have Sun Java System Directory Server's homephone attribute mapped to Active Directory's othertelephone attribute. You could use the Active Directory attributes drop-down list to change the mapping to homephone.

4 **You can also add or remove attribute values:**

- To add a value, enter the information in the New Value field and click Add.
- To remove a value, select the value from the list pane and click Remove.

5 **When you are done, click OK to apply your changes and close the Define Creation Mappings and Values dialog box.**

6 **Click OK again to close the Creation Mappings and Attributes dialog box.**

Removing Attributes

▼ To Remove Creation Attributes Mapping or Values

1 **Select the Object Creation tab, and click the Creation Attributes button located under the selected creation option.**

2 **When the Creation Mappings and Values dialog box is displayed, select the attribute from the table, and then click the Delete button.**

The attribute is removed from the table immediately.

3 **When you are done, click OK to close the Creation Mappings and Attributes dialog box.**

Specifying How Object Modifications Flow

Use the Attribute Modification tab to control how modifications made to user attributes and passwords will be propagated (flow) between your Sun and Windows systems.
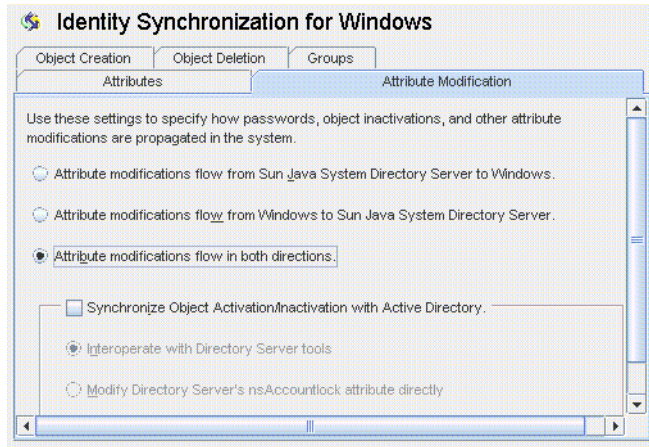
FIGURE 6–41   Attribute Modification Tab

You use this tab to configure the following:

- Specify the direction in which modifications flow between Directory Server and Windows directory sources.

- Control whether object activations and inactivations (*enables* and *disables* on Active Directory) will be synchronized between Directory Server and Active Directory directory sources, and specify the method in which user accounts are activated and inactivated.

Note – You cannot synchronize account statuses with Windows NT directory sources.

Specifying Direction

Select one of the following buttons to control how changes made in the Directory Server and Windows environments will be propagated between systems.

- **Attribute modifications flow from Sun Java System** *Directory Server to Windows*: Propagates changes made in the Directory Server environment to your Windows servers.

- **Attribute modifications flow from Windows to Sun Java System Directory Server** (*Default*): Propagates changes made in the Windows environment to your Directory Servers.

- **Attribute modifications flow in both directions** : Propagates changes bidirectionally (from one environment to the other environment).

Configuring and Synchronizing Object Activations and Inactivations

If you enable the Synchronize Object Activations/Inactivations with Active Directory box you can synchronize object activations and inactivations (known as *enables* and *disables* on Active Directory) between Directory Server and Active Directory directory sources.

Note – You cannot synchronize activations and inactivations with Windows NT directory sources.

▼ To Synchronize Object Activations/Inactivations:
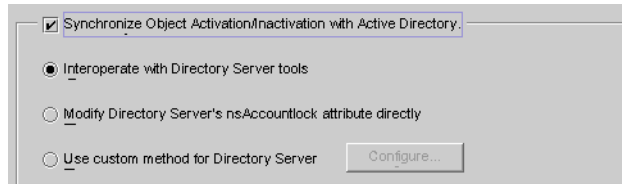
FIGURE 6–42 Synchronizing Object Activations and Inactivations

1 **Enable the Synchronize Object Inactivations between Directory Server & Active Directory box.**

2 **Enable one of the following buttons to specify how Identity Synchronization for Windows will detect and synchronize object activations and inactivations:**

- "Interoperating with Directory Server Tools" on page 178
- "Modifying Directory Server's NsAccountLock Attribute Directly" on page 179

Note – These options are mutually exclusive.

- "Using a Custom Method for Directory Server" on page 180

Interoperating with Directory Server Tools

Select this option if you use the Directory Server Console or command line tools to activate/inactivate an object. With this option selected Identity Synchronization for Windows cannot set or remove the nsAccountLock attribute directly. In addition, the program cannot detect objects that have been inactivated using other roles such as cn=nsdisabledrole, *database suffix* or roles that nest within other roles, such as cn=nsdisabledrole, *database suffix* or cn=nsmanageddisabledrole, *database suffix* .

- To activate objects, Identity Synchronization for Windows will remove the cn=nsmanageddisabledrole, *database suffix* value from the nsroledn attribute.

- To inactivate objects, Identity Synchronization for Windows will add the cn=nsmanageddisabledrole, *database suffix* value to the nsroledn attribute.

Note – If you enable the Interoperate with Directory Server Tools option, Identity Synchronization for Windows cannot set or remove the nsAccountLock attribute directly. In addition, Identity Synchronization for Windows cannot detect objects have been inactivated using other roles.

For example, cn=nsdisabledrole, *database suffix* or roles that nest within other roles such as cn=nsdisabledrole, *database suffix* or cn=nsmanageddisabledrole, *database suffix*.

Interoperating with Directory Server Tools describes how Identity Synchronization for Windows detects and synchronizes object activations/inactivations when you enable the Interoperate with Directory Server Tools option.

TABLE 6–1 Interoperating with Directory Server Tools

Activations	Inactivations
Identity Synchronization for Windows detects an activation only when the cn=nsmanageddisabledrole, *database suffix* role is removed from the object.	Identity Synchronization for Windows detects an inactivation only when the entry's nsroledn attribute includes the cn=nsmanageddisabledrole, *database suffix* role.
When synchronizing an object activation from Active Directory, Identity Synchronization for Windows activates the object by removing the cn=nsmanageddisabledrole, *database suffix* role from the object.	When synchronizing an object inactivation from Active Directory, Identity Synchronization for Windows inactivates the object by adding the cn=nsmanageddisabledrole, *database suffix* role to the object.

Modifying Directory Server's NsAccountLock Attribute Directly

Use this method when Directory Server activations and inactivations are based on Directory Server's operational attribute, nsAccountLock.

Note – When the Modify Directory Server's nsAccountLock attribute option is enabled, Identity Synchronization for Windows will not detect objects that are activated/inactivated using the Directory Server Console or command line utilities.

This attribute controls object states as follows:

- When nsAccountLock=true, the object is inactivated and the user cannot log in.
- When nsAccountLock=false (or has no value), the object is activated.

  Modifying Directory Server's NsAccountLock Attribute Directly describes how Identity Synchronization for Windows detects and synchronizes object activations/inactivations when you enable the Modify Directory Server's nsAccountLock Attribute Directly option.

TABLE 6–2   Modifying Directory Server's `nsAccountLock` Attribute Directly

Activation	Inactivation
Identity Synchronization for Windows detects an inactivated object only when the `nsAccountLock` attribute is set to **true**.	Identity Synchronization for Windows detects an activated object only when the `nsAccountLock` attribute is absent or set to **false**.
When synchronizing an object inactivationfrom Active Directory, Identity Synchronization for Windows removes the `nsAccountLock` attribute.	When synchronizing an object activation from Active Directory, Identity Synchronization for Windows sets the `nsAccountLock` attribute to **true**.

Using a Custom Method for Directory Server

Use this method when Directory Server activations and inactivations are controlled exclusively by an external application such as Sun Java System Access Manager (formerly Sun JES Identity Server).

When you configure a custom method for Directory Server, you must specify the following:

- How Identity Synchronization for Windows will detect that the external application has activated or inactivated an object in Directory Server.

- How Identity Synchronization for Windows will activate or inactivate the object when synchronizing from Active Directory to Directory Server.

Note – If you enable the Use custom method for Directory Server option, Identity Synchronization for Windows cannot lock objects out of the directory unless access to the directory is controlled by an external application, such as Access Manager.

To configure a Custom method for activations and inactivations, click the Configure button and the Configure Custom Method for Directory Server dialog box is displayed.
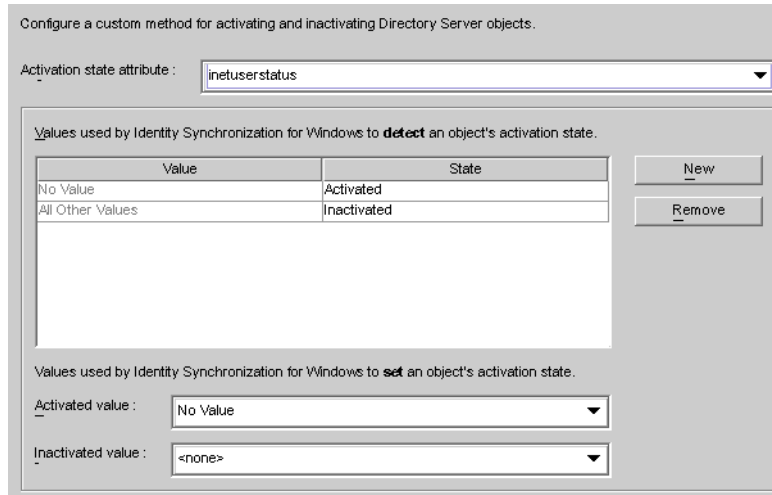
FIGURE 6–43 Configuring a Custom Method for Activations and Inactivations

This dialog contains the following features:

- **Activation state attribute drop-down list** : Use this list to specify an attribute that Identity Synchronization for Windows will use to synchronize activations and inactivations between Directory Server and Active Directory.

  The list contains all attributes in the schema for the currently selected Directory Server structural and auxiliary objectclasses.

- **Value and State table**: Use this table to specify when values associated with the selected attribute are activated or inactivated.

  - **Value column**: Use this column (in conjunction with the New and Remove buttons) to specify attribute values that will be used to indicate active or inactive states.

    The program automatically provides two values in this column:

    - **No Value**: Where the Activation state attribute has no value.
    - **All Other Values**: Where the Activation state attribute has a value, but that value is not specified in this Value and State table.

  - **State column**: Use this column to specify whether the Value entry (in the same row) corresponds to an object that is activated or inactivated.

Value	State	Result
No Value	Activated	If the attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as activated.

	Inactivated	If the attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as inactivated.
*user-defined***values**	Activated	If the attribute has the *user-defined* attribute, Identity Synchronization for Windows detects the object as activated.
	Inactivated	If the attribute has the user-defined attribute, Identity Synchronization for Windows detects the object as inactivated.
All Other Values	Activated	If the attribute has a value, but that value is not specified in the table, Identity Synchronization for Windows detects the object as activated.
	Inactivated	If the attribute has a value, but that value is not specified in the table, Identity Synchronization for Windows detects the object as inactivated.

- **New button**: Click this button to add new entries to the Value column.

- **Remove button**: Select an entry in the Value column, and then click this button to remove that entry.

- **Activated value and Inactivated value drop-down lists**: Use these two lists to specify values that Identity Synchronization for Windows will use to *set* an object's state.

   Synchronizing Activations and Inactivations

▼ **To Configure Identity Synchronization for Windows to Detect and Synchronize Object States between Directory Server and Active Directory**

1   Select an attribute from the Activation state attribute drop-down list.

2   Click the New button to add attribute values to the Value column of the table.

3   Click in the State column next to each of the Value entries and when the drop-down list is displayed, select Activated or Inactivated.
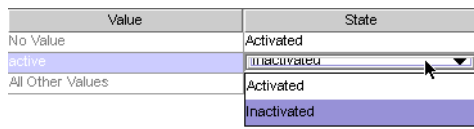
FIGURE 6–44   Selecting a State

For example, if you were using Access Manager:

4   Select the `inetuserstatus` attribute from the Activation state attribute drop-down list.

5   Click the New button and enter `active`, `inactive`, and `deleted` attribute values to the Value column of the table.

6   Click in the State column and select Activated or Inactivated for each value as follows:

- **No Value**: Activated
- **active**: Activated
- **inactive**: Inactivated
- **deleted**: Inactivated
- **All Other Values**: Inactivated

Based on this example, "Using a Custom Method for Directory Server" on page 180 describes how Identity Synchronization for Windows will detect and synchronize activations/inactivations when you enable the Use Custom Method for Directory Server option (using the `inetuserstatus` example).

Value	State	Result
No Value	Activated	If the `inetuserstatus` attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as activated.
active	Activated	If the attribute is **active** Identity Synchronization for Windows detects the object as activated.
inactive	Inactivated	If the attribute value is **inactive** Identity Synchronization for Windows detects the object as inactivated.
deleted	Inactivated	If the attribute value is **deleted** Identity Synchronization for Windows detects the object as inactivated.
All Other Values	Inactivated	If the attribute has a value, but that value is not specified in the table, Identity Synchronization for Windows detects the object as inactivated.

Setting Activations and Inactivations

As you populate the Value and State table with entries, Identity Synchronization for Windows automatically populates the **Activated value** and **Inactivated value** drop-down lists as follows:

- The Activated value list contains all values with an Activated status (for example **No Value** and **active**).

- The Inactivated value list contains all values with an Inactivated status (for example **inactive** and **deleted**).

- Neither list will contain the All Other Values value.

    Select a value from the Activated value and/or the Inactivated value drop-down lists to specify how Identity Synchronization for Windows will activate and/or inactivate an object when synchronizing from Active Directory.

- **Activated value**: Controls the object's active state.
  - **No Value**: If the object contains the active value, Identity Synchronization for Windows will set the state to activated in Directory Server.
  - **active**: If the object contains the active value, Identity Synchronization for Windows will set the state to activated in Directory Server.
- **Inactivated value**: Controls the object's active state.
  - **inactive** or **deleted**: Identity Synchronization for Windows will set the object's state to inactive in Directory Server.
  - *none*: Not a valid setting. You must select a value.

Note – You must specify an Inactivated value or your configuration will be invalid.

Using a Custom Method for Directory Server illustrates a completed Configure Custom Method for Directory Server dialog box.
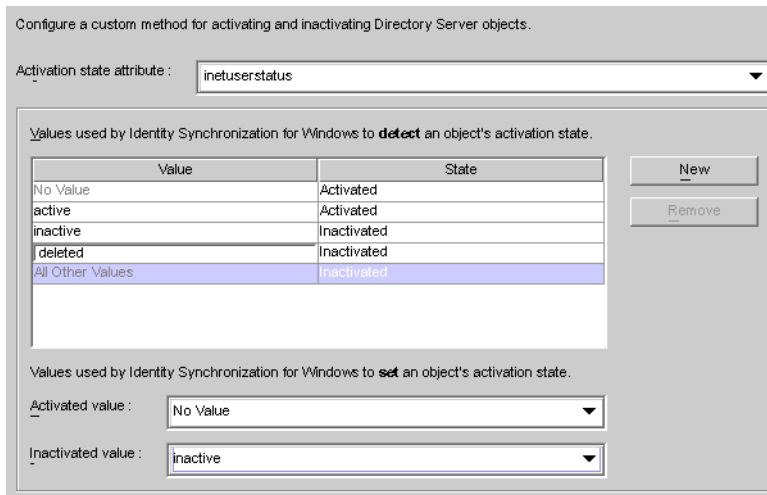
FIGURE 6–45   Example: Completed Dialog

Specifying Configuration Settings for Group Synchronization

If you enable Group Synchronization between Directory Server and Active Directory, you can synchronize the creation of groups, deletion of groups, and the membership changes within that group .

Note – Group Synchronization is not supported on Windows NT directory sources.

▼ To Synchronize Groups:

1   Under the Groups tab, select the Enable Group Synchronization check box.

2   Select one of the following Group Synchronization methods to specify how Identity
Synchronization for Windows will detect and synchronize various groups:

- *Domain Global Security*
- *Domain Global Distribution*
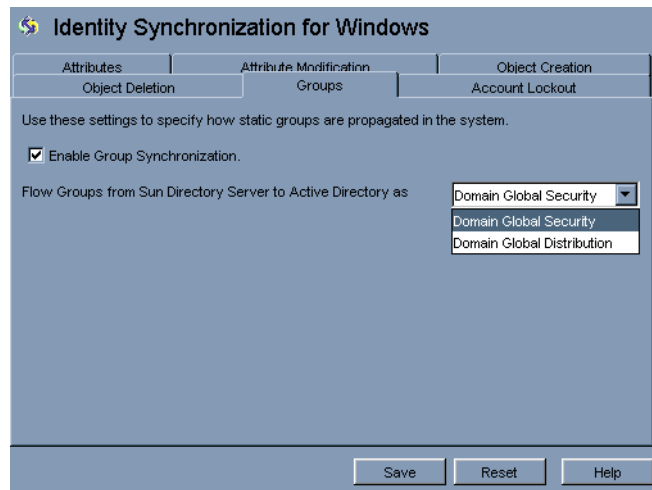
FIGURE 6–46   Enable Group Synchronization

Note – For more information about Domain Global Security, Domain Global Distribution, and
Active Directory; see the Microsoft Active Directory documentation.

Configure Identity Synchronization for Windows to Detect and Synchronize Groups Related Changes between Directory Server and Active Directory

You do not need to map any attribute manually for the group synchronization. When you press
Save, Identity Synchronization for Windows maps the attributes automatically.
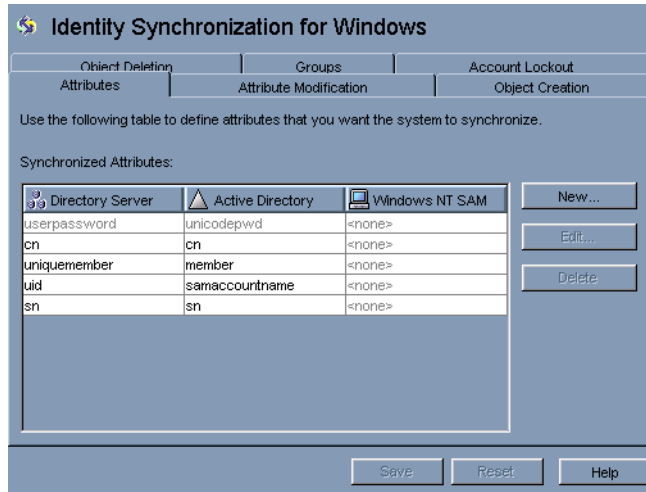
FIGURE 6–47   Attribute Mapping for Group Synchronization

Note –

1. Do not modify the mapping between the userpasswordand unicodepwd attributes.

2. To disable the group synchronization, deselect the Disable Group Synchronization check box.

3. Alternatively, you can enable or disable group synchronization using command line **idsync** *groupsync*. For more information, see Appendix A.

Configuring and Synchronizing Account Lockout and Unlockout

With this setting, you can synchronize account lockout and unlockout between Directory Server and Active Directory.

Note – Account lockout and unlockout synchronization is not supported on Windows NT directory servers.

Synchronize Account Lockout and Unlockout

Enable Account Lockout Synchronization between Directory Server & Active Directory box.
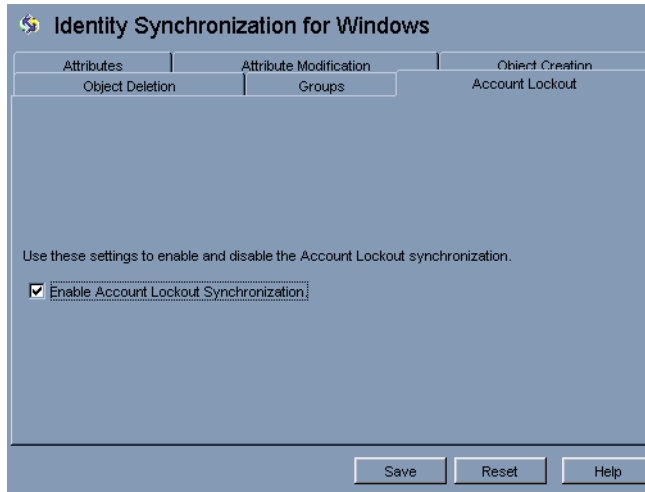
FIGURE 6–48   Enable Account Lockout Synchronization

Note – For Account Lockout and Unlockout to work correctly, it is recommended that you set the symmetric password policy at both ends. For example, if the password policy at Active Directory signifies a permanent lockout then the same password policy should be set at Directory Server.

Configure Identity Synchronization for Windows to Detect and Synchronize Account Lockout and Unlockout

You do not need to do any attribute mapping manually to achieve the account lockout and unlock synchronization. When you press Save, Identity Synchronization for Windows maps the attributes automatically.
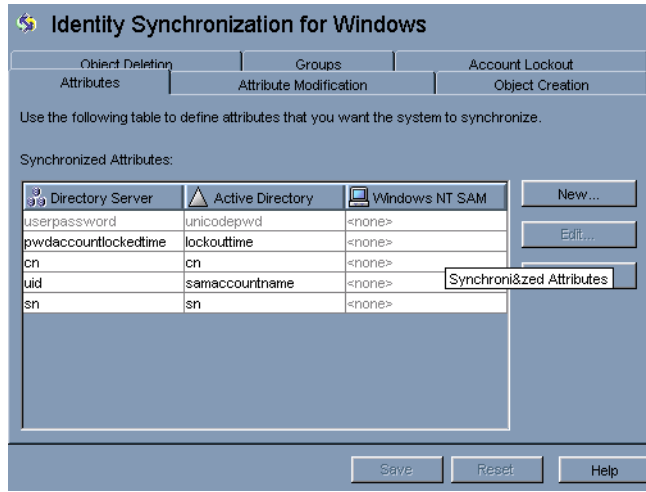
FIGURE 6–49 Attributes mapping for Account Lockout Synchronization

---

Note –

1. Do not modify the mapping between the `pwdaccountlockedtime`and `lockouttime`.

2. To disable the account lockout and unlockout synchronization, deselect the Enable Account Lockout Synchronization check box.

3. Alternatively, you can enable or disable the account lockout or unlockout synchronization using command line **idsync** *accountlockout*. For more information, see Appendix A.

Specifying How Deletions Flow

Use Object Deletions tab to specify how deleted user entries should flow between Directory Server and Active Directory systems.

Note – You cannot specify Object Deletions flow for Windows NT.

▼ To Specify how Deleted Entries Flow Between Directory Server and Active Directory Systems

1 Select the Identity Synchronization for Windows node at the top of the navigation pane, and then click the Object Deletion tab.
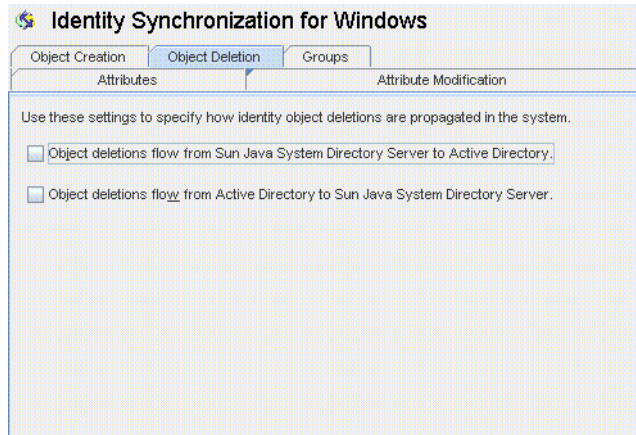
FIGURE 6–50   Propagating User Entry Deletions

2   **Enable or disable the flow of deletions as follows:**

- Enable **Object deletions flow from Sun Java System Directory Server to Active Directory** to propagate deletions from the Sun Directory Server environment to your Active Directory servers.

- Enable **Object deletions flow from Active Directory to Sun Java System Directory Server** to propagate deletions from the Active Directory environment to your Sun Directory Servers.

- Enable both options for bidirectional flow.

- Disable both options to prevent user deletions from propagating from one system to the other (*Default setting*).

Creating Synchronization User Lists

A Synchronization User List (SUL) specifies which users in Active Directory and Sun Directory Server will be synchronized. Every entry in the SUL passes through the Connector and is evaluated against the constraints you configured for that SUL.

Each SUL contains two elements, one to identify which Directory Server users to synchronize and one to identify which Windows users to synchronize.

Note – To synchronize users in a Directory Server with multiple Active Directory domains, you must define one SUL for each Active Directory domain.

For more information about defining and configuring SULs (including components of a definition, how to define multiple SULs, how multiple SULs are processed, and how to configure multiple Windows domain support) refer to Appendix D

Both of the SUL elements contain three definitions that identify which users to synchronize:

- **Base DN**: Location of the users to be synchronized (not applicable for NT)
- **Naming attribute**: Attribute used for newly created users (creation expression) (not applicable for NT)
- **Filter**: Excludes specified users from synchronization

▼ To Identify and Link User Types Between Servers

1 **Select the Synchronization User Lists node in the navigation tree, and then click New Synchronization User List button.**
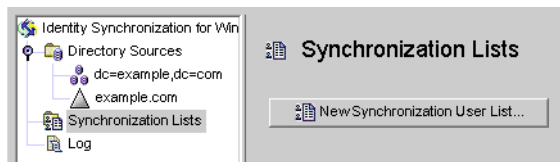
FIGURE 6–51   Creating a New Synchronization User List

The Define a Synchronization User List wizard is displayed.
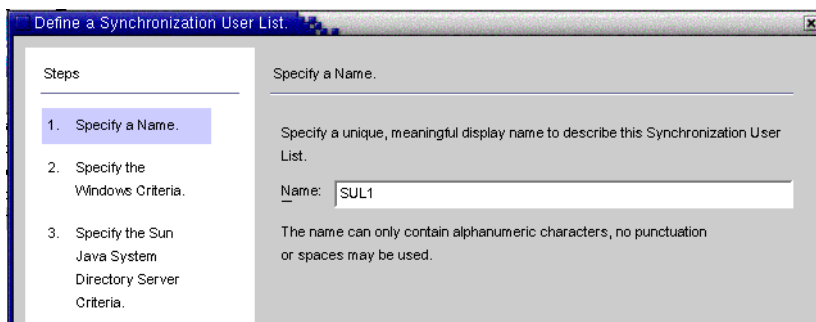
FIGURE 6–52   Specifying a Name for Your SUL

The program default for your first Synchronization User List is *SUL1*.

- If the default name is acceptable, click Next.
- If you want to use a different name, type a different name into the Name field and then click Next.
- Do not use spaces or any kind of punctuation in the SUL name.
- You must specify a name that is unique within the system.

   The Windows Criteria panel is displayed.

FIGURE 6–53   Specifying the Windows Criteria

2   Select a Windows Directory Source from the drop-down list.

Note – You cannot edit the Active Directory or Directory Server directory sources included in this SUL after you click the Finish button to create the SUL. When the Group Synchronization feature is enabled, the creation expression would be uid=%uid% or cn=%cn% in the Sun Java System Directory Server Criteria panel.

---

3   A *User Set Domain* **is the set of all the users to be synchronized. Enter the User Set Domain's Base DN, using one of the following methods:**

- Type the name into the text field (for example, **DC=example,DC=com**).
- Click the Browse button, to open the Set Base DN dialog box so you can look for, and select a Base DN.

   All users under the specified Base DN will be included in this SUL, unless you explicitly exclude them using a filter.

> **Note –** Base DNs and creation expressions are not allowed for Windows NT machines.
>
> You cannot edit the Active Directory or Directory Server directory sources included in this SUL after you click the Finish button to create the SUL. When the Group Synchronization feature is enabled, then the creation expression should be *uid=%uid%* in the Sun Java System Directory Server Criteria panel.

4   You can enter an equality, a presence, or a substring Filter to specify which users in this base DN
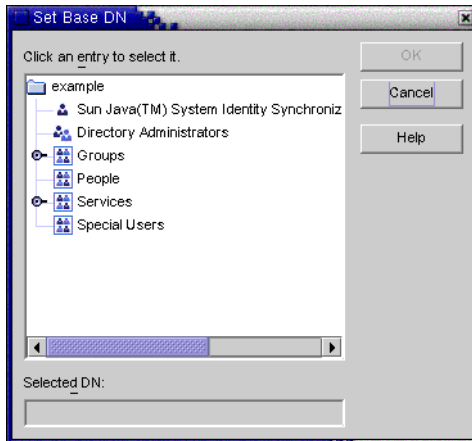
FIGURE 6–54   Selecting a Base DN

are synchronized. For example, if you are using the same base DN for multiple synchronization user lists, you may want to use a filter to distinguish between them.

The equality filter syntax is similar to LDAP query syntax, except that equality substrings allow *, &, |, =, ! characters only. For example, you can use the following filter to exclude the Administrator from your SUL:

```
(!(cn=Administrator))
```

The program should populate the Creation Expression field automatically.

> **Note –** A creation expression defines the parent DN and naming attribute used when new entries are propagated from Active Directory to Directory Server.
>
> A creation expression is not allowed for Sun directories unless you configured user attribute creations to flow from Active Directory to Directory Server. For more information, see "Specifying How Object Creations Flow" on page 171.

5   If the creation expression is missing or you want to change the existing entry, you can enter a creation expression for all Windows Active Directory synchronization user lists; for example:

`cn=%cn% ,cl=users,dc=example,dc=com`

If you are going to change the creation expression, you must select an attribute that you will be synchronizing. If necessary, go back to the Object Creation tab and use the Creation Attribute button to add and map this attribute.

6   Click Next to specify the Sun Java System Directory Server criteria.

7   When the Specify the Sun Java System Directory Server Criteria panel is displayed repeat Step 2 through Step 5 to provide the Directory Server criteria.
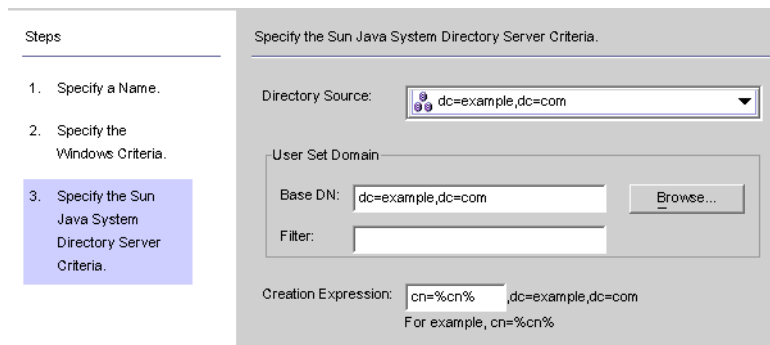
FIGURE 6–55   Specifying Directory Server Criteria

Note – You cannot edit the Active Directory or Directory Server directory sources included in this SUL after you click the Finish button to create the SUL.

8   When you are done, click Finish.

9   The program adds your new SUL node to the navigation tree and the Synchronization User List panel is displayed on the Configuration Tab.
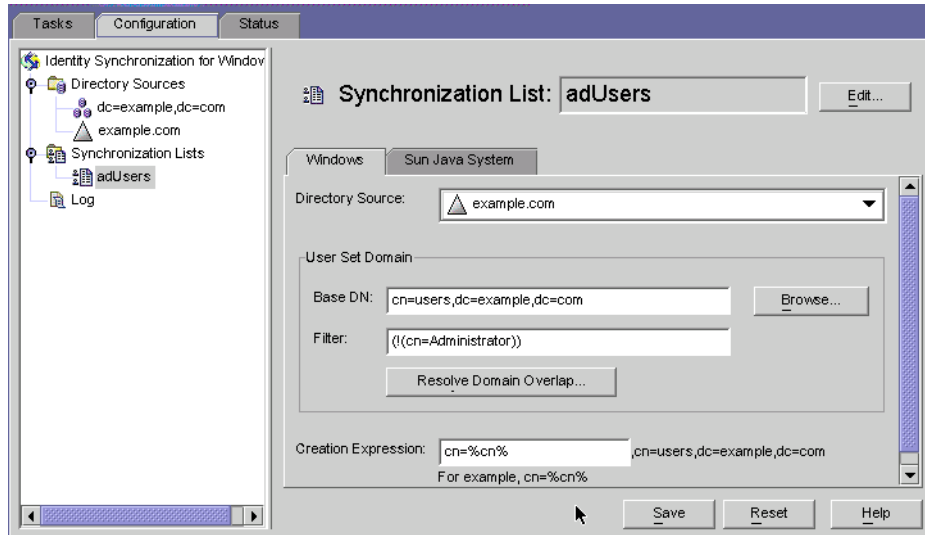
FIGURE 6–56 Synchronization List Panel

10 In cases where a user matches multiple lists, click the Resolve Domain Overlap button to define a preference for the synchronization user list.

11 Create a Synchronization User List that includes every directory source in your network except for the Directory Server.

Saving a Configuration

▼ To Save your Current Configuration from the Console Panels

1 Click Save to store your settings at this point.

2 The Configuration Validity Status window is displayed as the program evaluates your configuration settings.
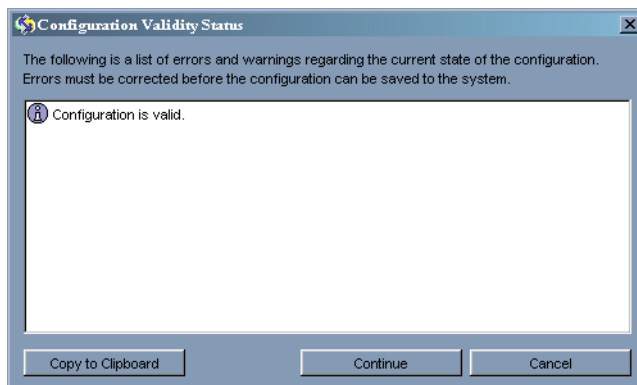
FIGURE 6–57 Configuration Validity Status Window

This panel confirms that your configuration is valid or identifies configuration problems that must be fixed.

Saving your configuration may take a few minutes because the program rewrites the information out to the configuration directory and notifies the system manager.

The system manager (a Core component) is responsible for distributing your configuration settings out to the components that need the information.

Note – Configuration validation errors are red and warnings are yellow.

- You cannot save a configuration with errors.
- You can save configurations with warnings, but it is better to try and clear the warnings first.

---

3 **If your configuration is valid, click Continue to save the configuration.**

A Connector Installation Instructions dialog box is displayed, giving instructions about how to proceed with installing the Identity Synchronization for Windows Connectors and subcomponents.

This list has now been updated with a To Do list that is customized for your deployment. (Up to this point, the steps were generic.) Note that you can also access and update the To Do list from the Status tab on the Identity Synchronization for Windows Console.
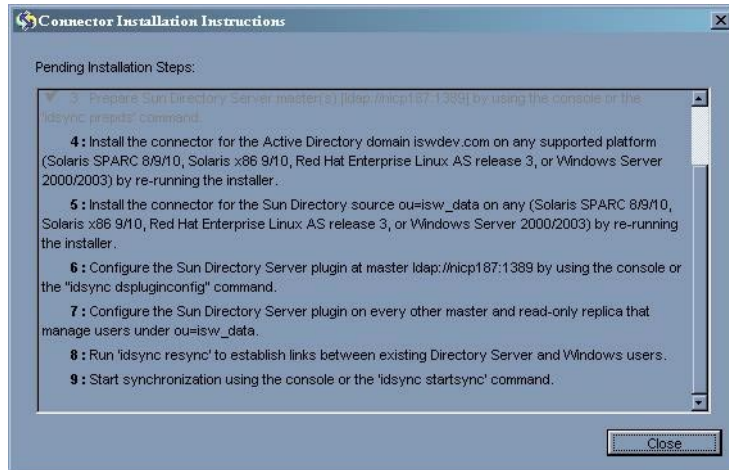
FIGURE 6–58   Instructions for Installing the Connectors

4   Read the information carefully and click OK.

After finishing the initial Core configuration, you are ready to install the Identity
Synchronization for Windows Connectors and subcomponents. Continue to Chapter 3 for
instructions.

7

Installing Connectors

This chapter provides instructions for installing the Identity Synchronization for Windows Connectors. The information is organized as follows:

Identity Synchronization for Windows uses Connectors to synchronize user passwords between directory sources, and uses subcomponents to enhance the Connector's change-detection and bidirectional synchronization support.

Before You Begin

Before starting the Connector configuring process, you should be aware of the following:

- Close the Console before starting the installation process. If the Console is open when you are installing a Connector, the program perceives a conflict about which component is adding configuration data to the server and generates an error message.

- Active Directory Connectors do not have subcomponents.

- Windows NT Connectors and subcomponents are installed simultaneously.

- You can install Directory Server or Active Directory Connectors on the same machine where you installed Core or you can install Connectors on another machine. (The Windows NT Connector must be installed on the Primary Domain Controller (PDC) of the domain being synchronized.)

- If you are installing the Connector on the same machine as Core, the program automatically installs the Connector in the same directory as Core.

- If you are installing the Connector on a different machine, the program will prompt you to specify the configuration directory information supplied during the Core installation.

You must run the installation program each time you install a Connector.

For example, if you are installing a Directory Server Connector and an Active Directory Connector, you will run the installation program twice after the Core is installed.

Running the Installation Program

Repeat the following steps each time you install a Connector.

▼ To Restart and Run the Installation Program

1 **Re-run the installation program on the machine where you want to install the Connector, as follows:**

- **On Solaris**: Change to the `installer` directory and then type **`./runInstaller.sh`** to execute the installation program.

- **On Linux**: Change to the `installer` directory and then type **`./runInstaller.sh`** to execute the installation program.

Note – To run the installation program in text-based mode, type **`./runInstaller.sh -nodisplay`**.

When you run the `runInstaller.sh` program, Identity Synchronization for Windows automatically masks passwords so they will not be echoed in the clear.

- **On Windows**: Change to the `installer` directory and then type **`setup.exe`** to execute the installation program.

2 **When the Welcome screen is displayed, read the information provided and then click Next to proceed to the Software License Agreement panel.**

3 **Read the license agreement, then select**

- **Yes (Accept License)** to accept the license terms and go to the next panel.
- **No** to stop the setup process and exit the installation program.

4 **The Sun Java System Directory Server panel is displayed. Specify the configuration directory location as follows:**

- **Configuration Directory Host**: Enter the fully qualified domain name (FQDN) of a Sun Java System Directory Server instance (affiliated with an Administration Server) where Identity Synchronization for Windows configuration information is stored. You must specify the same instance that you specified during the Core installation.

- **Configuration Directory Port** (*Defaults to port 389*): Specify a port for the configuration directory. You can leave the port set to the default or change to a different, available port.

  To enable SSL (Secure Socket Layer) between Core and the configuration directory, enable the Secure Port option and specify an SSL port (*default SSL port is 636*). Enabling this option prevents sensitive information from being passed in the clear over the network.

- **Configuration Root Suffix**: Select the root suffix that you specified during the Core installation from the menu. The Identity Synchronization for Windows configuration will be stored in this root suffix.

  Note – If the program could not detect a root suffix, and you enter the server information manually, you must click Refresh to repopulate the list of root suffixes.

  ---

5   **Click Next to open the Configuration Directory Credentials panel.**

6   **Enter the configuration directory Administrator's user ID and password.**

- If you specify admin as the user ID, you will not be required to specify the User ID as a DN.
- If you use any other user ID, then you must specify the ID as a full DN. For example, *cn=Directory Manager*.

  Note – These credentials will be sent without encryption unless you enabled SSL in.

  ---

7   **Click Next to open the Configuration Password panel where you must enter the configuration password you specified when you installed Core.**

    Also, if Core has not been installed on this machine, you will be prompted to provide the location of the Java Home directory (see "Installing Core" on page 132).

8   **When you are finished, click Next.**

  Note – At this point, the installation process becomes specific to the type of Connector you are installing.

Installing Connectors

This section explains how to install the three types of Identity Synchronization for Windows Connectors, as follows

> **Note –** You are not required to install Connectors in any particular order, but do not attempt to install any Connectors simultaneously.

Installing the Directory Server Connector

After completing the steps described in
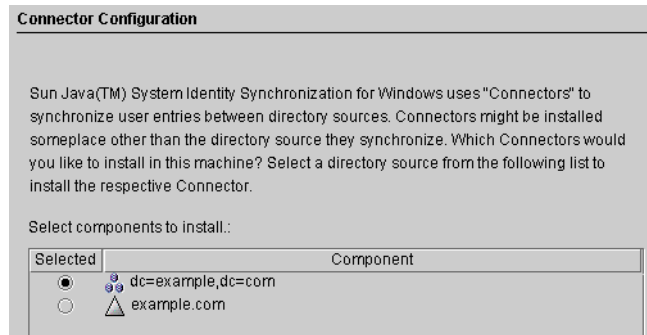
FIGURE 7–1   Selecting the Directory Server Connector

The Select components to install list contains only those Connector components that have not yet been installed. For example, after you install the Directory Server Connector (`dc=example,dc=com`), the program will remove the entry from the list pane.

The following table contains some example directory source entries.

TABLE 7–1   Directory Source Examples

Directory Source	Example Entry
Sun Java System Directory Server	`dc=example,dc=com`
Windows Active Directory	`example.com`
Windows NT SAM	`EXAMPLE`

▼ To Install the Directory Server Connector

1   **Enable the button next to the Directory Server Connector component and then click Next.**

The Directory Server Connector Credentials panel is displayed.

Directory Server Connector Credentials

Enter the directory manager credentials for the Sun Java(TM) System Directory Server(s) associated with the connector being installed.

Primary: ldap://machine1.example.com:389

Primary Directory Server User DN: cn=Directory Manager

Primary Directory Server Password:

Secondary: none

Secondary Directory Server User DN: cn=Directory Manager

Secondary Directory Server Password:

Note – The program automatically completes the User DN fields with your fully qualified Directory Manager distinguished name, but you can change the information if necessary.

Enter the following information:

- **Primary Directory Server User DN**: If necessary, change the default user DN by entering a fully qualified Directory Manager distinguished name.

- **Primary Directory Server Password**: Enter your Directory Manager password.

  If you are using a secondary master, the Secondary Directory Server User Name and Password fields will be active. The program automatically completes the Directory Manager DN field with the same entries provided for the Primary Directory Server User DN and Password fields. You can change this information if necessary.

  The program will verify that the Directory Server was prepared and ready to synchronize data. When you prepared Directory Server (), the program creates an account that the Connector will use to connect to Directory Server (for example, uid=PSWConnector,*suffix*).

2 Click Next to proceed to the Connector Port Configuration pane.

Connector Port Configuration

Some Sun Java(TM) System Identity Synchronization for Windows Connectors require a TCP/IP port number. You must specify a TCP/IP server port number to enable communication between the Connector and its subcomponent(s). You must specify a port number that is not being used by any other applications on this machine.

Fully Qualified Local Host Name: machine1.example.com

Connector Port Number:

3 **Enter the Fully Qualified Local Host Name with the domain and an available port number where the Connector will listen. (Specifying a port already in use will result in an error message.)**

4 **Click Next and the Ready to Install pane is displayed to provide information about the Connector's installation location and how much disk space is required for the installation. When you are ready, click the Install Now button.**
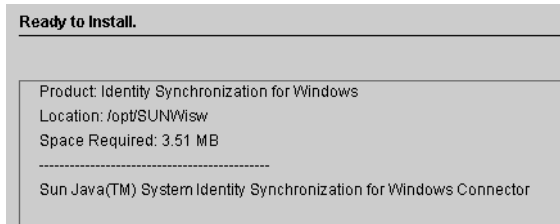
Note – If you installed Core on the local machine, the Ready to Install pane will indicate that zero space is required to install the Connector. This situation occurs because the Core installation has already installed the Connector binaries. Because there are no additional binaries to install, no additional space is required.

If you are installing the Connector on a machine other than where you installed Core, then the Ready to Install pane will indicate how much space is required to complete the Connector installation on the local machine.

The Connector installation is accomplished in two steps:

- An Installing pane is displayed, with a progress bar, while the program installs the binaries.
- Next, the Component Configuration pane displays a progress bar. This step takes several minutes to complete.

Note – If you did not close the Console before starting the installation, the following warning displays ("Installing the Directory Server Connector" on page 200). Click Reset in the Console to reload the Connector's configuration settings.

When both steps are complete, an Installation Summary pane is displayed.
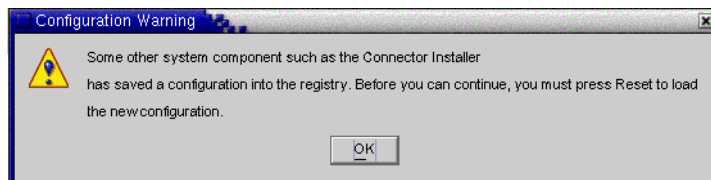
Note – Directory Server plugin gets configured for preferred and secondary hosts (if any).
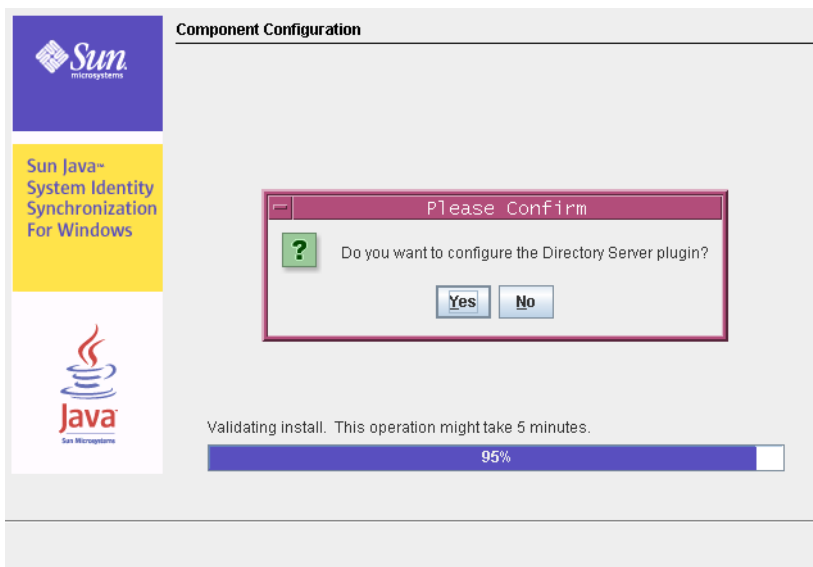
Note –

a. Clicking Yes configures the Directory Server plugin in all the hosts (preferred and secondary).

b. Clicking No enables you to configure the plugin later using command line **idsync** *dspluginconfig*. For more information, see Appendix A.

5 Click the Details button if you want to review the installation log.

- **On Solaris**: Installation logs are written to /var/sadm/install/logs/

- **On Linux**: Installation logs are written to /var/sadm/install/logs/

- **On Windows**: Installation logs are written to the %TEMP% directory, which is usually a subdirectory of the Local Settings folder located underC:\Documents and Settings\Administrator

> **Note –** On some Windows systems (such as Windows 2000 Advanced Server), the Local
> Settings folder is a hidden folder.
>
> To view this folder and the Temp subdirectory, open your Windows Explorer and select
> Tools → Folder Options from the menu bar. When the Folder Options dialog box is
> displayed, select the View tab and enable the Show Hidden Files option.

6   Click Next to display the "To Do list" panel, which shows the list of successfully completed and pending steps.

This is a list of remaining installation and configuration steps:

✔ 1 : Install the Identity Synchronization core components.
✔ 2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importcnf'.
✔ 3 : Prepare Sun Directory Server master(s) [ldap://isw080:1389] by using the console or the 'idsync prepds' command.
✔ 4 : Install the connector for the Sun Directory source ou=isw_data on any (Solaris SPARC 8/9/10, Solaris x86 9/10, Red Hat Enterprise Linux AS release 3, or Windows Server 2000/2003) by re-running the installer.
✔ 5 : Configure the Sun Directory Server plugin at master ldap://isw080:1389 by using the console or the "idsync dspluginconfig" command.
6 : Install the connector for the Active Directory domain metaqa.com on any supported platform (Solaris SPARC 8/9/10, Solaris x86 9/10, Red Hat Enterprise Linux AS release 3, or Windows Server 2000/2003) by re-running the installer.
7 : Configure the Sun Directory Server plugin on every other master and read-only replica that manage users under ou=isw_data.
8 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.
9 : Start synchronization using the console or the 'idsync startsync' command.

7   When you are done with the panel, click Finished.

After installing the Directory Server Connector, you can install other Connectors that you configured when you configured the resources (Chapter 6):

- Install additional Directory Server Connectors: Restart the installation program (using the instructions in "Running the Installation Program" on page 198) and then repeat Step 1 through Step 7.

- Install an Active Directory Connector: Go to "Installing an Active Directory Connector" on page 205.

- Install a Windows NT Connector: Go to "Installing the Windows NT Connector" on page 208.

Configuring Identity Synchronization for Windows Plug-in when Chained Suffix exists

This configuration is needed only when the chained suffix exists in the Directory Server instance where Identity Synchronization for Windows Plug-in is installed. If Identity

Synchronization for Windows Plug-in is not configured to search on chained suffix, MODIFY and BIND operations performed on the Directory Server where the Identity Synchronization for Windows Plug-in is installed, will fail.

In the Directory Server instance where the chained suffix is created, perform the following operations:

Execute the following LDIF script using ldapmodify utility:

```
dn: cn=config,cn=chaining database,cn=plugins,cn=config
changetype: modify
add: nspossiblechainingcomponents
nspossiblechainingcomponents: cn=pswsync,cn=plugins,cn=config
```

You can perform the similar operation by using the following procedure:

1. Select the Configuration tab.
2. Click the Data node that displays in the left pane.
3. Select the Chaining tab in the right pane.
4. Add Identity Synchronization for Windows Plug-in (cn=pswsync,cn=plugins,cn=config) to the components that are allowed to chain.
5. Save the changes and exit.

Installing an Active Directory Connector

After you install the Directory Server Connector and if you have other configured Connectors to install, the installation program will give you the option of installing the Connectors before you see the Connector Configuration pane.

Connector Configuration

Sun Java(TM) System Identity Synchronization for Windows uses "Connectors" to synchronize user entries between directory sources. Connectors might be installed someplace other than the directory source they synchronize. Which Connectors would you like to install in this machine? Select a directory source from the following list to install the respective Connector.

Select components to install.:

Selected	Component
◉	⚠ metaqa.com

FIGURE 7–2   Selecting the Connector

The component list contains only those Connector components that have not yet been installed. For example, if you already installed the Directory Server Connector (dc=example,dc=com in this case), it will not be listed.

▼ To Install an Active Directory Connector

1   Enable the Connector button and click Next.

The Connector Configuration panel displays.
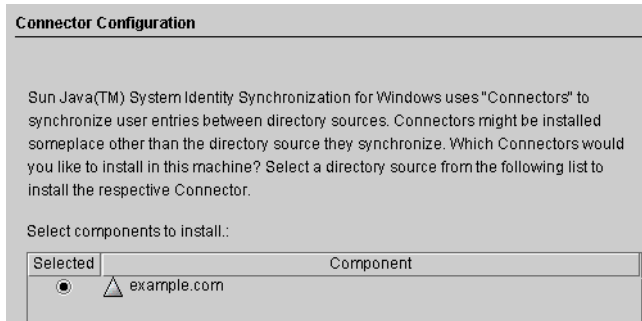
The Select components to install list contains only those Connector components that have not yet been installed. For example, after you install the Directory Server Connector (dc=example,dc=com in this case), the program will remove the entry from this list pane.

2   Enable the button next to the Active Directory component and then click Next.

The Ready to Install pane is displayed to provide information about the Connector's installation location and how much disk space is required for the installation.

Note – If you installed Core on the local machine, the Ready to Install pane will indicate that zero space is required to install the Connector. This situation occurs because the Core installation has already installed the Connector binaries. Because there are no additional binaries to install, no additional space is required.

If you are installing the Connector on a machine other than where you installed Core, then the Ready to Install pane will indicate how much space is required to complete the Connector installation on the local machine.

3   When you are ready, click the Install Now button.

An Installing pane is displayed, with a progress bar, while the program installs the binaries, and then an Installation Summary pane is displayed to confirm the installation is finished.

4   Click the Details button if you want to review the installation log.

- **On Solaris**: Installation logs are written to /var/sadm/install/logs/

- **On Linux**: Installation logs are written to /var/sadm/install/logs/

- **On Windows**: Installation logs are written to the %TEMP% directory, which is a subdirectory of the Local Settings folder located underC:\Documents and Settings\Administrator

---

Note – On some Windows systems (such as Windows 2000 Advanced Server), the Local Settings folder is a hidden folder.

To view this folder and the Temp subdirectory, open your Windows Explorer and select Tools → Folder Options from the menu bar. When the Folder Options dialog box is displayed, select the View tab and enable the Show Hidden Files option.

---

5   Click Next to display the "To Do list" panel, which shows the list of successfully completed and pending steps.

6   When you are done with the panel, click Finished to exit the installation program.

After installing the Active Directory Connector, you can install other Connectors that you configured when you configured resources (Chapter 6):

- Install additional Active Directory Connectors: Restart the installation program (see "Running the Installation Program" on page 198) and then repeat through.

- Install a Windows NT Connector: Go to "Installing the Windows NT Connector" on page 208.
- Install additional Directory Server Connectors: Restart the installation program (using the instructions in "Running the Installation Program" on page 198) and then repeat Step 1 through Step 6.

Installing the Windows NT Connector

You must install the Windows NT Connector on the Primary Domain Controller (PDC) of the domain you configured.

▼ To Install a Windows NT Connector and the NT subcomponents

1 **Enable the Windows NT Connector button and click Next.**

2 **When the Connector Port Configuration pane is displayed, enter the Fully Qualified Local Host Name with the domain and an available port number where the Connector will listen. (Specifying a port already in use will result in an error message.)**

3 **When you are done, click Next.**
   The Ready to Install pane is displayed to provide information about the Connector's installation location and how much disk space is required.

4 **When you are ready, click the Install Now button.**
   The Connector installation is accomplished in two steps:

   - An Installing pane is displayed, with a progress bar, while the program installs the binaries.
   - Next, the Component Configuration pane displays a progress bar. This step takes several minutes to complete.

   Note – If you did not close the Console before starting the installation, a warning displays (see "Installing the Directory Server Connector" on page 200). Click Reset in the Console to reload the Connector's configuration settings.

   When both steps are complete, an Installation Summary pane is displayed.

5 **Click the Details button if you want to review the installation log.**
   Installation logs are written to the %TEMP% directory, which is C:\TEMP on most Windows NT systems.

6   Click Close to exit the installation program.

After installing the Windows NT Connector, you can install other Connectors that you configured when you configured resources (Chapter 6):

- To install additional Windows NT Connectors, restart the installation program. For more information, see "Running the Installation Program" on page 198 and then repeat Step 1 through Step 6.
- To install Directory Server Connector, refer to "Installing the Directory Server Connector" on page 200.
- To install Active Directory Connector, refer to "Installing an Active Directory Connector" on page 205.

8

Synchronizing Existing Users and User Groups

The Identity Synchronization for Windows command line utility provides the `idsync resync` subcommand to bootstrap deployments with existing users or groups. This command uses administrator-specified matching rules to link existing entries, to populate an empty directory with the contents of a remote directory, or to bulk-synchronize attribute values (including passwords) between two existing user and group populations.

This chapter explains how to use the `idsync resync` subcommand and synchronize existing users and groups for new Identity Synchronization for Windows installations. In addition, this chapter provides instructions for starting and stopping synchronization and services. The information is organized as follows:

Note – You must finish installing Core and the Connectors before trying to synchronize existing users.

For more information about the `idsync resync` subcommand, see Appendix A

Synchronizing Existing Users and User Groups summarizes the post-installation steps to follow based on existing user and group populations:

Post-Installation Steps Based on Existing User and Group Populations

TABLE 8–1   Post-Installation Steps Based on Existing User Populations

Users Exist In		Post-Installation Steps	
Windows	**Directory Server**	**Synchronize Existing Users**	**Do NOT Synchronize Existing Users**
No	No	None	None
No	Yes	Run `idsync resync -o Sun -c` to create existing Directory Server users in Windows.	None
Yes	No	Run `idsync resync -c` to create existing Windows users in Directory Server.	Run `idsync resync -u` to populate the connector's local cache of user entries.
Yes	Yes	Run `idsync resync -f <filename> -k` to link the users only, and then run `idsync resync -o Sun` to resynchronize existing users from Directory Server.	Run `idsync resync -u` to populate the connector's local cache of user entries.

Note – If Group Synchronization is enabled then the groups are synchronized in the same way as the users are synchronized.

Using idsync resync

This section explains the synchronizing processes, describes the proper syntax for using the `idsync resync` subcommand, and explains how to verify that the processes completed successfully. The information is organized as follows:

- "Resynchronizing Users or Groups" on page 212
- "Linking Users" on page 213
- "idsync resync Options" on page 214
- "Checking Results in the Central Log" on page 217

Resynchronizing Users or Groups

You need to resynchronize the user entries when two directory sources become out of sync. Use the `idsync resync` command to create users, user groups, and synchronize user and user group

attributes in two directory sources. Specifically, you can use the idsync resync command to populate an empty Directory Server with the existing Active Directory or Windows NT SAM domain users.

The idsync resync command can be used in any of the following ways:

- If there are users that exist on Directory Server and Windows, you must run the idsync resync command to synchronize those users.

- If you do not want to synchronize existing users to Directory Server, then run idsync resync with the -u argument, which updates the object cache only and does not synchronize the Windows' entries to Directory Server.

- If you have existing Windows users and do not run idsync resync, then changes to these users may or may not be propagated; and depending on flow settings, these users might even be automatically created in Directory Server. You must run idsync resync again, even if you have already run the command.

Note – You cannot use the idsync resync command to synchronize passwords (except to invalidate Directory Server passwords to force on-demand password synchronization in an Active Directory environment).

When the Group Synchronization feature is enabled, both the users as well as the groups associated with the users are synchronized between the data sources configured. No additional options are required while using the resync command for Group Synchronization.

Linking Users

After populating Active Directory and Directory Server with users and installing the Active Directory and Directory Server Connectors (before starting synchronization), you must use the idsync resync command to ensure that all existing users are *linked* in the two directory sources.

What is *linking*? Identity Synchronization for Windows correlates the same user on Directory Server and on Windows by storing the following unique, immutable identifiers:

- The dspswuserlink attribute of each Directory Server user entry
- The objectguid attribute for each Active Directory user
- A combination of the domain name and the RID for each Windows NT SAM user

Storing this immutable identifier allows Identity Synchronization for Windows to synchronize other key identifiers, such as uid and cn. The dspswuserlink attribute is populated when:

- Identity Synchronization for Windows creates a new user in Directory Server (after a new user is synchronized from Windows or by runningidsync resync -c)

- Identity Synchronization for Windows creates a new user on Windows (after synchronizing a new user from Directory Server or by running `idsync resync -c -o Sun`)

- You run `idsync resync -c -f` to link entries that already exist on Directory Server and Windows as described in this chapter.

To link existing users, you must provide rules for matching users between the two directories. For example, to link a user entry in two directories, both the first names and last names must match in both directory entries.

Linking user entries and resolving data conflicts could be described as more art than science. There are many reasons why the `idsync resync` subcommand might fail to link two users in opposing directory sources and depends to a large extent on the consistency of the data in the linked directories.

One strategy for using `idsync resync` is to use the `-n` argument, which runs the operation in "*safe mode*" so you can preview the effects of an operation with no actual changes. Running in safe mode allows you to refine the linking criteria gradually until you find an optimum set of user matching criteria.

However, you should be aware that there is a balance to be achieved through linkage accuracy and linkage coverage.

For example, if both directory sources contain an employee ID or social security number, you might begin with linking criteria that includes this number only. You might think that to improve linkage accuracy, you should include a last name attribute in the criteria as well. However, you could lose linkages because entries that would have matched on ID alone did not match because there were inconsistent last name values in the data. You will have to go through a data cleansing process for entries that fail to link.

Note – If Group Synchronization is enabled then the groups are linked in the same way as the users are linked.

---

idsync resync Options

The `idsync resync` command accepts the following options.

TABLE 8–2 idsync resync Usage

Argument	Meaning
-a *<ldap-filter>*	Specifies an LDAP filter to limit the entries to be synchronizedThe filter will be applied to the source of the resynchronization operation. For example, if you specify idsync resync -o Sun -a "usid=*" all Directory Server users that have a uid attribute will be synchronized to Active Directory.
-l *<sul-to-sync>*	Specifies individual Synchronization User Lists (SULs) to resynchronize **Note**: You can specify multiple SUL IDs to resynchronize multiple SULs or, if you do not specify any SUL IDs, the program will resynchronize all of your SULs.
-o (Sun | Windows)	Specifies the source of the resynchronization operation ■ **Sun**: Sets attribute values for Windows entries to corresponding attribute values in Sun Java System Directory Server directory source entries. ■ **Windows**: Sets attribute values for Sun Java System Directory Server entries to corresponding attribute values in Windows directory source entries. (*Default is Windows.*)
-c	Creates a user entry automatically if the corresponding user is not found at destination ■ Randomly generates a cryptographically secure password for users created in Active Directory or Windows NT. ■ Automatically creates a special password value ({PSWSYNC} ***INVALID PASSWORD***) for users created in Directory Server (unless you specify the -i option) **Note**: Identity Synchronization for Windows will attempt to create users even if you have not configured creations in that direction. For example, if you have not configured Identity Synchronization for Windows to synchronize from Windows to Sun (or vice versa), but you specify the -c argument, Identity Synchronization for Windows will try to create users that are not found.
-i (ALL_USERS | NEW_USERS |)	Resets passwords for user entries synchronized in a Sun directory source, forcing password synchronization within the current domain for those users the next time the user password is required. ■ **ALL_USERS**: Forces on-demand password synchronization for all synchronized users ■ **NEW_USERS**: Forces on-demand password synchronization for newly created users only

TABLE 8–2 idsync resync Usage   *(Continued)*

Argument	Meaning
-u	Updates the object cache.
	This argument updates the local cache of user entries for a Windows directory source only, which prevents pre-existing Windows users from being created in Directory Server. If you use this argument, Windows user entries are not synchronized with Directory Server user entries. This argument is valid only when the resync source is Windows.
-x	Deletes all destination user entries that do not match a source entry.
-n	Runs in safe mode so you can preview the effects of an operation with no actual changes.

TABLE 8–3 Will idsync resync invalidate the user's password on Directory Server?

	User has an entry on Active Directory and on Directory Server that is linked.	User has an entry on Active Directory and on Directory Server that are not linked.	User has an entry on Active Directory, but not on Directory Server.
-i ALL_USERS	Yes	Yes	Yes
-i NEW_USERS	No	No	Yes
No -i value	No	No	No

The following table provides examples to illustrate the results of combining different arguments (The – h, -p, -D, -w, -, and -s arguments are defaulted and have been omitted for brevity).

TABLE 8–4 idsync resync Usage Samples

Arguments	Result
idsync resync	
	Displays a resync **usage statement.**
idsync resync -i ALL_USERS	Invalidates the passwords of all users to force on-demand password synchronization (valid in Active Directory environments only).
	In mixed environments (with both Active Directory and NT domains), you must explicitly list Active Directory SULs.
idsync resync -c -i NEW_USERS	Creates users that are not found on Directory Server and invalidates their passwords to force on-demand password synchronization. Use this command to populate an empty Directory Server instance with existing Windows users.

TABLE 8–4 idsync resync Usage Samples   *(Continued)*

Arguments	Result
idsync resync	
	Displays a resync **usage statement.**
idsync resync -c -l SUL_sales -l SUL_finance	Creates all existing Active Directory users on Directory Server for the SUL_sales and SUL_finance SULs only (but does not force on-demand password synchronization).
idsync resync -n	Runs in safe mode so you can preview the effects of the resync operation with no actual changes.
idsync resync -o Sun -a "(sn=Smith)"	Synchronizes all Directory Server users with the last name (sn) Smith, on Windows.
idsync resync -u	Updates the object cache for Windows Connectors only to prevent existing users from being created in Directory Server. No users are actually synchronized.
idsync resync -f link.cfg	Links unlinked users based on linking criteria specified in the link.cfg file. Identity Synchronization for Windows does not create or modify users, but the Directory Server passwords of newly linked users will be set to the Active Directory users' passwords.

Note – When you use idsync resync to link users, be aware that you should use indexed attributes for the operation. Non-indexed attributes can affect performance.

If there are multiple attributes in the UserMatchingCriteria set, and at least one of them is indexed, then performance will probably be acceptable. However, if there no indexed attributes in the UserMatchingCriteria, then performance will be unacceptable with a large directory.

Checking Results in the Central Log

The results of all idsync resync operations are reported in a special central log named resync.log. This log lists all of the users that were properly linked and synchronized, those that failed to link, and those that were previously linked.

Note – Some pre-existing special Active Directory users (such as Administrator and Guest) might appear in this log as failures.

Starting and Stopping Synchronization

Starting and stopping synchronization *does not* start or stop individual Java processes, daemons, or services. Once you begin synchronization, stopping synchronization only pauses the operation. When you restart synchronization, the program resumes synchronization from where it stopped and no changes will be lost.

▼ To Start or Stop Synchronization

1. **In the Sun Java System Server Console navigation pane, select the Identity Synchronization for Windows instance.**

2. **When the Identity Synchronization for Windows pane is displayed, click the Open button in the upper right corner.**

3. **When you are prompted, enter the configuration password.**

4. **Select the Tasks tab.**
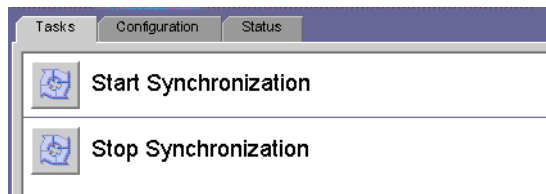
   FIGURE 8–1   Starting and Stopping Synchronization

   - To start synchronization, click Start Synchronization.
   - To stop synchronization, click Stop Synchronization.

   Note – You can also start and stop synchronization using the idsync startsync and idsync stopsync command line utilities. For detailed instructions, see

Resynchronized Users/Groups

To resynchronize groups, the Group Synchronization feature must be enabled either through the console or through the command line interface.

To know about how to enable the Group Synchronization feature, see "Specifying Configuration Settings for Group Synchronization" on page 184

Starting and Stopping Services

Identity Synchronization for Windows and Message Queue are installed as *daemons* on Solaris and Linux, and as *services* on Windows. These processes start automatically when the system boots, but you can also start and stop them manually, as follows:

- **On** *Solaris*: From the command line,
    - Enter /etc/init.d/isw start to start all Identity Synchronization for Windows processes.
    - Enter /etc/init.d/isw stop to stop all Identity Synchronization for Windows processes.
    - Enter /etc/init.d/imq start to start the Message Queue broker.
    - Enter /etc/init.d/imq stop to stop the Message Queue broker.
- **On** *Linux*: From the command line,
    - Enter /etc/init.d/isw start to start all Identity Synchronization for Windows processes.
    - Enter /etc/init.d/isw stop to stop all Identity Synchronization for Windows processes.
    - Enter /etc/init.d/imq start to start the Message Queue broker.
    - Enter /etc/init.d/imq stop to stop the Message Queue broker.
- **On Windows**:
    - From the Windows Start menu:
        1. Select Start → Settings → Control Panel → Administrative Services.
        2. When the Administrative Services dialog box is displayed, double-click the Services icon to open the Services dialog box.
        3. Select Identity Synchronization for Windows and then select Action → Start (or Stop) from the menu bar. Repeat for iMQ Broker.
    - From the command line, enter the net command to control the services.

Note – Pause 30 seconds after stopping the Identity Synchronization for Windows daemon/service before starting it again. Connectors can take several seconds to cleanly shut themselves down.

9

Removing the Software

This section contains procedures for removing Identity Synchronization for Windows 6.0 in the following sections:

Planning for Uninstallation

Before removing the software keep in mind the following points:

- You must uninstall subcomponents and the Directory Server Plug-in before you uninstall their associated connectors, and uninstall all the connectors before Core. (The Active Directory Connector does not have any subcomponents to uninstall.)

  Failure to uninstall one of these components in the proper order will prevent you from selecting and uninstalling the other components. For example, if you do not uninstall the connectors first, you cannot select Core for uninstallation.

- You must uninstall the Directory Server Plug-in before you uninstall Core.

  Uninstalling Core first will remove the Plug-in bits without unregistering them from the Directory Server, which prevents the Directory Server from starting unless you manually remove cn=pswsync,cn=plugins,cn=config.

- In replicated environments with replicas (in addition to primary and secondary servers) you must uninstall the Directory Server Plug-in and then restart the servers.

- The order in which you uninstall connectors does not matter.

- After uninstalling a Sun Java System Directory Server or Windows Connector, you must perform some additional steps to reinstall the Connector on a different machine or to use a different server port.

In this case, you must uninstall and reinstall all of the corresponding subcomponents, and restart the Identity Synchronization for Windows daemon/service where Core is installed (see "Starting and Stopping Services" on page 219.

- You must run the uninstall.cmd script (located in the isw-*hostname* directory) on Windows 2000 and NT platforms. (You must run this batch file as Administrator.)

- You must run the runUninstall.sh script (located in the installation directory, /opt/SUN/isw, by default) on the Solaris or Linux operating systems. (You must run this script as root.)

Note – You must follow the instructions for uninstalling product components and subcomponents *explicitly*, and verify that you have uninstalled all components successfully.

Uninstalling the Software

Your system may contain any or all of the following Identity Synchronization for Windows components:

- Active Directory Connectors
- Directory Server Connectors and Plug-ins
- Core

  Your Windows NT system may contain the Windows NT Connector and subcomponents.

  Use runUninstaller.sh (Solaris or Linux) or uninstall.cmd (Windows) to remove all connectors and subcomponents and then remove Core (if installed).

This section provides instructions for the following:

- "Uninstalling Connectors" on page 222
- "To Uninstall Core" on page 224

Uninstalling Connectors

▼ To Uninstall the Connectors

1 Start the uninstaller program (runUninstaller.sh **on Solaris or on Linux or** uninstall.cmd **on Windows).**

These programs are located in the installation directory (which is the /opt/SUNWisw directory by default).

2 At the Welcome screen click Next.

3 **Enter the Configuration Directory Host name and Port number.**

- Select the root suffix of the configuration directory. (If necessary, click Refresh to see the list of suffixes.)

- For secure communication between the uninstall program and the configuration directory server, enable the Secure Port box and specify the Directory Server's SSL port number.

4 **Enter your administrator's name and password for the configuration directory.**

5 **Select the connector(s) to be uninstalled.**

Note – The selected connectors *must* be present on the target host.

---

6 **Click Next to perform further uninstallation related tasks.**

7 **A summary window appears. Please follow the instructions presented in this window.**

- **On Solaris systems**: Uninstallation logs are written to /var/sadm/install/logs/

- **On Linux systems**: Uninstallation logs are written to /var/sadm/install/logs/

- **On Windows systems**: Uninstallation logs are written to the %TEMP% directory, which is a subdirectory of the Local Settings folder located in

  C:\Documents and Settings\Administrator

Note – On some Windows systems such as Windows 2000 Advanced Server, the Local Settings folder is a hidden folder. To view this folder and the Temp subdirectory:

Open your Windows Explorer and select Tools → Folder Options from the menu bar. When the Folder Options dialog box is displayed, select the View tab and enable the Show Hidden Files option.

---

8 **Click Close to exit the program.**

9 **If there are no other connectors installed on the target host, then you can safely remove the** isw-*hostname* **folder.**

10 **Repeat for all hosts where connectors are installed.**

▼ To Uninstall Core

Note – You must uninstall the Directory Server Plug-in before you uninstall Core.

Uninstalling Core before the Plug-in removes the Plug-in bits without unregistering them from the Directory Server, which will prevent the Directory Server from starting unless you manually remove cn=pswsync,cn=plugins,cn=config.

Use the following instructions to uninstall Core:

1 **Start the uninstaller program:**

- **On Windows machines:**

  a. **Click** Start, **and then choose** Settings → Control Panel.

  b. **Double-click** Add/Remove Programs.

  c. **In the** Add/Remove Programs **window, select** Identity Synchronization for Windows, **then click** Remove.

- **On Solaris or Linux machines, execute** runUninstaller.sh **or** uninstall.cmd **on Windows.**
  These programs are located in the installation directory (which is the /opt/SUNWisw directory on Solaris and /opt/sun/isw directory on Linux by default).

2 **In the Welcome screen click Next.**

3 **Enter the Configuration Directory Host name and Port number.**

  a. **Select the root suffix of the configuration directory. (If necessary, click Refresh to see the list of suffixes.)**

  b. **For secure communication between the uninstall program and the configuration directory server, enable the Secure Port box and specify the Directory Server's SSL port number.**

4 **Enter your administrator's name and password for the configuration directory.**

5 **Select Core to be uninstalled and click Next.**

6 **Enter the configuration directory URL, click Refresh, and select the appropriate root suffix from the drop-down list.**

7 **Click Next to perform further uninstallation related tasks.**

8 **A summary window appears. Please follow the instructions presented in this window.**

   a. **On Solaris systems: Uninstallation logs are written to** `/var/sadm/install/logs/`

   b. **On Linux systems: Uninstallation logs are written to** `/var/sadm/install/logs/`

   c. **On Windows systems: Uninstallation logs are written to the** `%TEMP%` **directory, which is a subdirectory of the** `Local Settings` **folder located under**

   `C:\Documents and Settings\Administrator`

   Note – On some Windows systems (such as Windows 2000 Advanced Server), the `Local Settings` folder is a hidden folder.

   To view this folder and the `Temp` subdirectory:

   Open your Windows Explorer and select Tools → Folder Options from the menu bar. When the Folder Options dialog box is displayed, select the View tab and enable the Show Hidden Files option.

   ---

9 **Click Close to exit the program.**

Note – If you are unable to run the connector uninstaller for a given connector for any reason (for example, if you lost the connector files during a hard drive failure), use the idsync resetconn subcommand (see).

This command resets the connector state in the configuration directory to *uninstalled* so that you can reinstall it elsewhere. The resetconn subcommand is similar to other commands that access the configuration directory, and it provides two options:

- **-e** *dir-source*: Specifies the name of the directory source to be reset. (Connectors are identified in the installers by their directory source name.)
- **-n** (safe mode): Indicates whether the arguments specified for the command are correct without doing any work.

Example command:

```
idsync resetconn -D "cn=Directory Manager"-w [-h CR-hostname]
[-p 389] [-s dc=example,dc=sun,dc=com] -q [-Z] [-P "cert8.db"]
 [-m "secmod.db"] -e "dc=central, dc=example,dc=com" [-n]
```

resetconn Output:

```
NOTICE: This program will reset the installation state
to UNINSTALLED for the Connector associated with the
specified DirectorySource 'dc=central,dc=example,dc=com'.
Changing the Connector to an UNINSTALLED state is a
last resort. This is NOT meant to be used for
uninstalling connectors.It is typically used if you
lost a machine with the connector on it and can not
run the uninstaller.  Additionally, this program will
rewrite the existing configuration. This can be a
lengthy process. Before proceeding, you should stop
the Console, any running installers, and all other
system processes. You may want to export the ou=Services
tree in the configuration directory to ldif as a backup.
Do you want to reset the installer settings
for the connector (y/n)?
```

Uninstalling the Console Manually

After you have removed all other Identity Synchronization for Windows components, you may have to manually uninstall the Console.

From Solaris or Linux Systems

▼ To Uninstall the Console from Solaris or Linux

1   Delete the following subtree from the configuration directory:

```
cn=Sun Java (TM) System Identity Synchronization for Windows,
cn=server_group,cn=hostname,
ou=domain_ name, o=netscaperoot
```

2   For all console installations, remove all of the `.jar` **files with an** *isw* **prefix from the following directory:**

serverroot/*server*/`java/jars`

From Windows Systems

▼ To Uninstall the Console from a Windows Active Directory or NT system

1   Delete the following subtree from the configuration directory:

```
cn=Sun Java (TM) System Identity Synchronization for Windows,
cn=server_group, cn=hostname,
ou=domain_name, o=netscaperoot
```

2   For all console installations, remove all of the `.jar` **files with an** *isw* **prefix from the following directory:**

serverroot/*server*/`java/jars`

10

Configuring Security

This chapter provides important information about configuring security for your deployment. The information is organized as follows:

Note – This chapter assumes that you are familiar with the basic concepts of public-key cryptography and Secure Sockets Layer (SSL) protocol, and that you understand the concepts of intranet, extranet, Internet security, and the role of digital certificates in an enterprise. If you are new to these concepts, please refer to the security-related appendixes of the *Managing Servers with iPlanet Console 5.0* manual.

Security Overview

Passwords are sensitive information; therefore, Identity Synchronization for Windows takes security precautions to ensure that user and administrative password credentials used to access the directories being synchronized are not compromised.

This section covers the following security methodologies:

- "Message Queue Access Controls" on page 233
- "Directory Credentials" on page 234
- "Persistent Storage Protection Summary" on page 234

This security approach aims to prevent the following events from taking place:

- An eavesdropper intercepting a clear text password over the network

- An attacker manipulating a connector to change a user's password to a value of their choosing, which is equivalent to capturing the user's clear text password

- An attacker gaining access to a privileged component of Identity Synchronization for Windows

- An unprivileged user recovering a password from a file stored on disk.

- An intruder recovering a password from a hard disk that was removed from one of the components of the system. This could be a password being synchronized, or it could be a system password that is used to access a directory.

Specifying a Configuration Password

To protect sensitive information while it is stored in the product's configuration directory and while it is transferred over the network, Identity Synchronization for Windows uses a *configuration password*. You (the administrator) specify a configuration password when you install Core, and you must provide this password when you open the Console or run the Identity Synchronization for Windows installation program.

Note – The system manager must access the configuration password before passing it to the connector; consequently, the system manager stores this password in its initialization file.

File system access controls prevent non-privileged users from accessing the system manager's initialization file. The Identity Synchronization for Windows installation program does not enforce a password policy for this password.

To increase security when you select a configuration password, see "Hardening Your Security" on page 235.

Using SSL

You can configure Identity Synchronization for Windows to use LDAP over SSL everywhere that components use LDAP. All access to Message Queue is protected with SSL.

You must use SSL between the Active Directory Connector and Active Directory when you are synchronizing from Directory Server to Active Directory.

Requiring Trusted SSL Certificates

By default, connectors configured to use SSL will accept any SSL certificate that the server (i.e.Directory Server or Active Directory) returns — which includes untrusted, expired, and invalid certificates. All network traffic between the connector and server will be encrypted, but the connector will not detect a server that is impersonating the true Active Directory or Directory Server.

To force the connector to accept only trusted certificates, use the Console to enable the Require trusted SSL certificates option on the Specify Advanced Security Options panel of the Directory Source Configuration wizard (see "Creating an Active Directory Source" on page 156). After enabling this option, you must add the appropriate CA certificates to the connector's certificate database as reported by idsync certinfo.

Generated 3DES Keys

A 3DES key generated from the configuration password is used to secure all sensitive information in the product's configuration directory. With the exception of log messages, all messages to the Message Queue are encrypted with per-topic 3DES keys. Messages sent between connectors and subcomponents are encrypted with per session 3DES keys. The Directory Server Plug-in encrypts all user password changes with a 3DES key.

SSL and 3DES Keys Protection Summary

"SSL and 3DES Keys Protection Summary" on page 231 summarizes how Identity Synchronization for Windows protects sensitive information that is sent over the network.

TABLE 10–1   Protecting Sensitive Information Using Network Security

Use this Protection Method	Between the Following Information Types:
LDAP over SSL (optional)	■ Directory Server Connector and Directory Server, Active Directory Connector and Active Directory ■ Directory Server Plug-in and Active Directory ■ Command line interfaces and the product's configuration directory ■ Console and the product's configuration directory ■ Console and Active Directory Global Catalog ■ Console and Active Directory domains or Directory Servers being synchronized ■ Message Queue broker and the product's configuration directory ■ Connectors, system manager, central logger, command line interfaces, and Console may authenticate the Message Queue over LDAPS ■ Installer and the Configuration Directory Server ■ Installer and Active Directory ■ Installer and the Directory Server being synchronized
Encrypted with 3DES keys (default)	■ Directory Server Connector and Directory Server Plug-in (all data) ■ Windows NT Connector, Windows NT Password Filter DLL, and Windows NT Change Detector (all data) ■ All sensitive information in the product's configuration directory ■ All messages sent between connectors and subcomponents (encrypted with per-session 3DES keys) ■ All (non-log) messages sent over Message Queue

"SSL and 3DES Keys Protection Summary" on page 231 contains an overview of the security features discussed in this section.
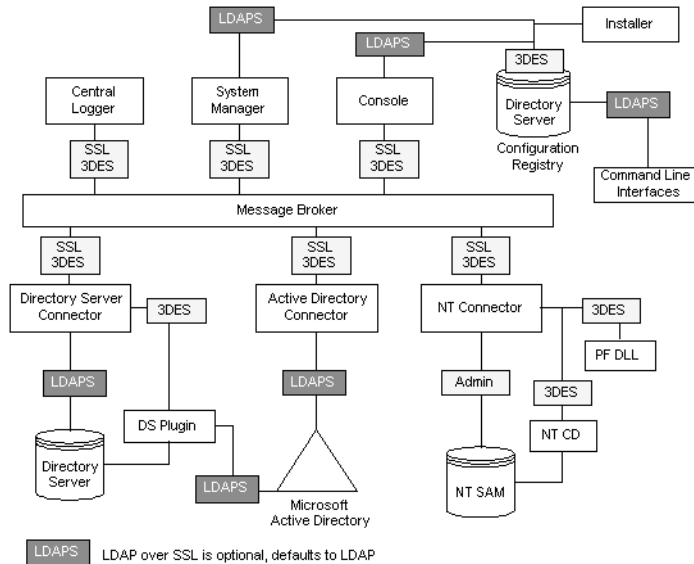
FIGURE 10–1 Security Overview for Identity Synchronization for Windows

Message Queue Access Controls

Identity Synchronization for Windows uses Message Queue's access control to prevent unauthorized access to message subscription and publishing, allowing each connector to trust messages that it receives.

Unique username and passwords known only to Message Queue and to the connector are provided to access the Message Queue broker. Each message sent over the Message Queue is encrypted with a per topic 3DES key, which protects the message contents and prevents outsiders who do not know the topic key from sending meaningful messages. These measures prevent (a) an attacker from sending falsified password synchronization messages to connectors and (b) an attacker from impersonating a connector and receiving actual password updates.

Note – By default, clients of the Message Queue, such as the connectors and system manager, accept any SSL certificate that the Message Queue broker returns. See "Hardening Your Security" on page 235 for more information to enhance Message Queue certificate validation and other Message Queue-related security issues.

Directory Credentials

Privileged credentials are required by the connectors to change passwords in Active Directory and the Directory Servers being synchronized. These privileged credentials are encrypted before they are stored in the product's configuration directory.

Persistent Storage Protection Summary

"Persistent Storage Protection Summary" on page 234 summarize how Identity Synchronization for Windows protects sensitive information that is stored on disk.

TABLE 10–2   Persistent Storage Protection

Persistent Storage	Confidential Information	Protection
Product's Configuration Stored in a Configuration Directory Server	Credentials for accessing the directories and per Message Queue topic 3DES keys are stored in the product's configuration directory.	All sensitive information stored in the product's configuration directory is encrypted with a 3DES key that is generated from the configuration password. See "Hardening Your Security" on page 235 for recommendations to further protect the product's configuration directory.
Directory Server Retro Changelog	The Directory Server Plug-in captures password changes and encrypts them before writing them to the Directory Server Retro Changelog.	The Directory Server Plug-in encrypts all user password changes with a 3DES key that is unique to each deployment.
Message Queue Broker Persistent Storage	The Message Queue broker stores password synchronization messages sent between all connectors.	With the exception of log messages, all persisted messages are encrypted with per-topic 3DES keys.
Message Queue Broker Directory Credentials	The Message Queue broker authenticates users against the product's configuration directory. It connects to the configuration directory using the directory administrative user name and password provided during Core installation.	The directory password is stored in a passfile, which is protected with file system access controls.
System Manager Boot File	The system manager's boot file contains information for accessing the configuration. This includes the configuration password and the directory administrative user name and password provided during Core installation.	This file is protected with file system access controls.
Connectors and Central Logger Boot Files	Each connector as well as the central logger have an initial configuration file with credentials for accessing the Message Queue.	These files are protected with file system access controls.

TABLE 10–2 Persistent Storage Protection   *(Continued)*

Persistent Storage	Confidential Information	Protection
Directory Server Plug-in Boot Configuration	The Plug-in's configuration, stored in cn=config, includes credentials for connecting to the connector.	The cn=config subtree is protected with ACIs and the dse.ldif file, which mirrors this tree, is protected with file system access controls.
NT Password Filter DLL and NT Change Detector Boot Configuration	The NT subcomponent's configuration, which is stored in the Windows registry, includes credentials for connecting to the connector.	If access to the PDCs registry is not secure, these registry keys can be protected with access controls.
Windows Connector's Object Cache	Windows connectors store hashed user passwords in the connector's object cache.	The passwords are not stored in the clear but encrypted with MD5 hashes. These database files are protected with file system access controls.(see "Hardening Your Security" on page 235

Hardening Your Security

This section depicts potential security weaknesses in the current release of the product and recommendations as to how to extend and harden security outside the product's default configuration. It includes the following:

- "Configuration Password" on page 235
- "Creating Configuration Directory Credentials" on page 235
- "Message Queue Client Certificate Validation" on page 236
- "Message Queue Self-Signed SSL Certificate" on page 237
- "Access to the Message Queue Broker" on page 237
- "Configuration Directory Certificate Validation" on page 237
- "Restricting Access to the Configuration Directory" on page 237

Configuration Password

The configuration password is used to protect sensitive configuration information but the installation program does not enforce any password policy for this password; be sure that this password follows some strict guidelines choose a complex password that is not easily guessed and follow standard policy guidelines for strong passwords.

For example, it should be at least eight characters long, include upper case letters, lower case letters, and non-alphanumeric characters. It should not include your name, initials, or dates.

Creating Configuration Directory Credentials

To access the Directory Server where the product's configuration directory resides, your credentials must be in the Configuration Administrators group. However, if you need to create credentials other than *admin* for any reason, consider the following:

The installation program requires you to provide credentials for a user stored in the Console administrative subtree. However, the Core installation program will not expand users other than *admin* into "uid=admin,ou=Administrators, ou=TopologyManagement, o=NetscapeRoot". Therefore, you must specify the entire DN during Core installation.

▼ To Create a New User Other Than *admin*

1 Create a user in:

ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

2 Add the new credentials to the Configuration Administrators group.

3 Set ACIs to allow only this user or all users in the Configuration Administrators group to access the Directory Server where the product's configuration directory is stored.

4 Specify entire DN during Core installation.

For more information about managing access controls in the Directory Server, see Chapter 6, "Directory Server Access Control," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*

Message Queue Client Certificate Validation

By default, clients of the Message Queue, such as the connectors and system manager, accept any SSL certificate that the Message Queue broker returns.

▼ To Validate the Message Queue Client Certificate

1 To override this setting and force Message Queue clients to validate the Message Queue broker's certificate, edit:

installation_root/resources/WatchList.properties

2 Add the following to the JVM arguments of each process in Watchlist.properties:

-Djavax.net.ssl.trustStore=*keystore_path* -DimqSSLIsHostTrusted=false

3 Restart the Identity Synchronization for Windows daemon or service.

The javax.net.ssl.trustStore property should point to a JSEE keystore that trusts the broker certificate, for example, /etc/imq/keystore can be used on the machine where Core was installed because this is the same keystore used by the broker.

Message Queue Self-Signed SSL Certificate

By default, the Message Queue broker uses a self-signed SSL certificate. To install a different certificate, use the `keytool` utility that ships with Java to modify the broker's keystore (`/var/imq/instances/isw-broker/etc/keystore` on Solaris, `/var/opt/sun/mq/instances/isw-broker/etc/keystore` on Linux, and *mq_installation_root* `/var/instances/isw-broker/etc/keystore` on Windows 2000). The alias of the certificate must be `imq`.

Access to the Message Queue Broker

By default, the Message Queue uses dynamic ports for all services except for its port mapper. To access the broker trough a firewall or restrict the set of hosts that can connect to the broker, the broker should use fixed ports for all services.

This can be achieved by setting the `imq.`*service_name protocol_type* `.port` broker configuration properties. Refer to the *Sun Java System Message Queue Administration Guide* for more information.

Configuration Directory Certificate Validation

The system manager accepts any certificate when connecting to the product's configuration directory over SSL; the Message Queue broker accepts any certificate when connecting to the product's configuration directory over SSL. Currently, there is no way to make either the system manager or the Message Queue broker validate the product's configuration directory SSL certificates.

Restricting Access to the Configuration Directory

When Core is installed, the process of adding information to the Directory Server where the product's configuration directory is stored does not include adding any access control information. To restrict access to only configuration Administrators, the following ACI can be used:

```
(targetattr = "*")
(target = "ldap://ou=IdentitySynchronization,
ou=Services,dc=example,dc=com")
(version 3.0;acl "Test";deny (all)
(groupdn != "ldap://cn=Configuration Administrators,
ou=Groups, ou=TopologyManagement, o=NetscapeRoot");)
```

For more information about managing access controls in the Directory Server, see Chapter 6, "Directory Server Access Control," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*

Securing Replicated Configurations

Deployments connecting to Directory Servers using replication follow the same rules identified in "Security Overview" on page 229. This section gives an example replicated configuration and explains how to enable use of SSL in this configuration.

Note – For an overview of planning, deploying, and securing replicated configurations see
Appendix D

"Securing Replicated Configurations" on page 238 lists the configuration components requiring CA certificates and identifies which certificates are required where.

TABLE 10–3   MMR Configuration Components Requiring CA Certificates

Component	Required CA certificates
Preferred Directory Server Replicated Master	Active Directory System
Secondary Directory Server Replicated Master	Active Directory System
Read-only Directory Server Hub(s)	Preferred Directory Server Replicated Master
	Secondary Directory Server Replicated Master
Directory Server Connector	Preferred Directory Server Replicated Master
	Secondary Directory Server Replicated Master
Active Directory Connector	Active Directory System

Replicated configuration shows Identity Synchronization for Windows installed in an MMR configuration, where there are two replicated Directory Server masters with multiple Directory Server read-only hubs or consumers. Each Directory Server has a Plug-in and there is only one Directory Server Connector, one Active Directory system, and one Active Directory Connector.
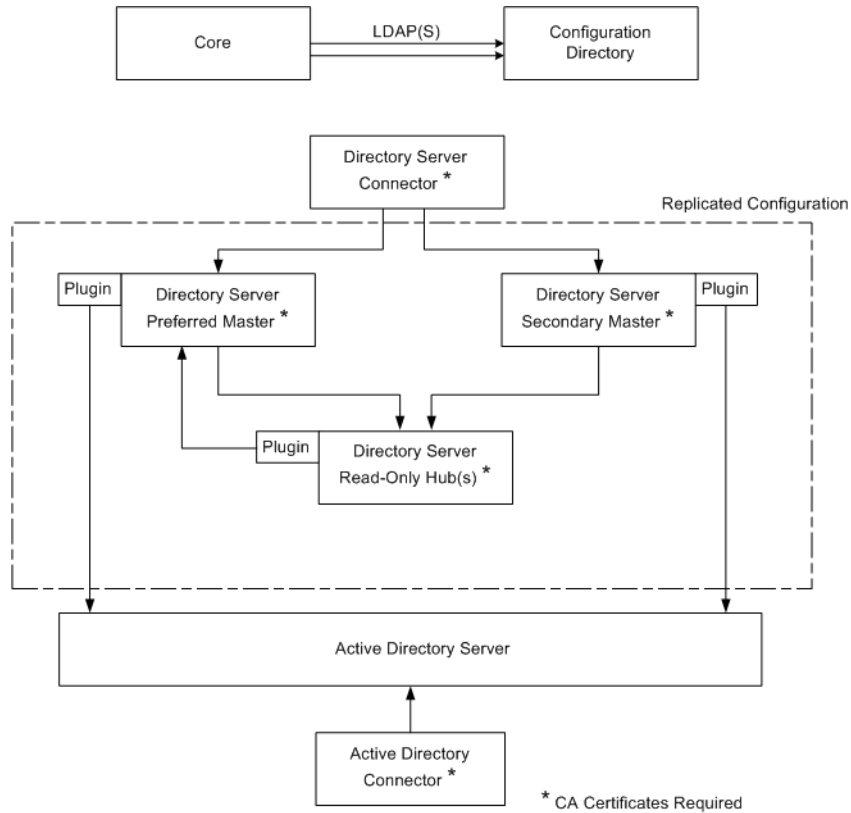
FIGURE 10–2 Replicated Configuration

When the Directory Server source is configured for SSL, you must make sure that both the preferred and secondary Directory Server certificates are trusted by the replica Directory Server. This is true for every Directory Server Plug-in of type other that you install on a system with a Directory Server hub or read-only replica.

Note – Directory Server Plug-ins have access to the same CA certificates as its associated Directory Server.

The above diagram is specific to two Directory Server masters. But you can extended this to contain multiple masters.

Using idsync certinfo

Use the idsync certinfo utility to determine what certificates are required based on the current Identity Synchronization for Windows SSL settings. Execute idsync certinfo to retrieve information about what certificates are required in each certificate database.

Note – You must be sure that when you are configuring the Directory Server source for SSL, both the preferred and secondary Directory Server source certificates are trusted by the replica Directory Server for all Directory subcomponents or Plug-ins.

If Identity Synchronization for Windows tries to establish SSL connections (with the trust all certificates setting enabled), and the server's hostname does not match the hostname provided in the certificate presented by the server during the SSL negotiation phase, the Identity Synchronization for Windows Connector will refuse to establish the connection.

The directory source hostname in the Identity Synchronization for Windows configuration must always match the hostname embedded in the certificate used by that directory source.

Arguments

Arguments describes the arguments you can use with the idsync certinfo subcommand.

TABLE 10–4 certinfo Arguments

Argument	Description
-h *CR-hostname*	Specifies the configuration directory hostname. This argument defaults to the values specified during Core installation.
-p *CR-port-no*	Specifies the configuration directory LDAP port number. (*Default is 389.*)
-D *bind-DN*	Specifies the configuration directory bind distinguished name (DN). This argument defaults to the values specified during Core installation.
-w *bind-password* | -	Specifies the configuration directory bind password. The - value reads the password from standard input (STDIN).
-s *rootsuffix*	Specifies the configuration directory rootsuffix. Where rootsuffix is a distinguished name such as dc=example,dc=com.This argument defaults to the values specified during Core installation.
-q *configuration_password*	Specifies the configuration password. The - value reads the password from standard input (STDIN).

Usage

The following example uses idsync certinfo to search for system components designated to run under SSL communications. The results of this example identifies two connectors (CNN101 and CNN100) and provides instructions as to where to import the appropriate CA certificate.

```
:\Program Files\Sun\MPS\isw-
hostname\bin idsync certinfo -h
CR-hostname -p 389 -D
"cn=Directory Manager" -w dirmanager -s dc=example,dc=com
 -q password
Connector: CNN101
Certificate Database Location: C:\Program Files\Sun\MPS\isw-
hostname\etc\CNN101
Get 'Active Directory CA' certificate from Active Directory
and import into Active Directory Connector certificate db
for server ldaps::/
hostname.example.com:636
Connector: CNN100 Certificate Database Location:
C:\Program Files\Sun\MPS\isw-
hostname\etc\CNN100
Export 'Directory Server CA' certificate
from Directory Server certificate db and
import into Directory Server Connector certificate db
ldaps://hostname.example.com:636
Export 'Active Directory CA' certificate
from Active Directory Server
hostname.example.sun.com:389
and import into Directory Server Server certificate db
for server ldaps://hostname.example.com:638
SUCCESS
```

Enabling SSL in Directory Server

Follow these steps to enable SSL in a Directory Server using a self-signed certificate.

> **Note –** These abbreviated procedures are for your convenience. Refer to the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide* for more information.
>
> - On Windows, use the `certutil`version bundled with Identity Synchronization for Windows 6.0 within the *ISW-host-name*\shared\bin folder.
> - On Solaris, `certutil` is installed in `/usr/sfw/bin` by default.
> - On Linux, `certutil` is installed in `/opt/sun/private/bin` by default.

▼ To Enable SSL in Directory Server

Note: The following procedure is applicable only for Solaris because Directory Server 6.0 is not yet available for Windows

1 Create a DS instance

/opt/SUNWdsee/ds6/bin/dsadm create -p *non-ldap-port*-P *ldap-secure-port*
<*DS-server-root*>/slapd-<*hostname*>

2 Start the instance

/opt/SUNWdsee/ds6/bin/dsadm start <*DS-server-root*>/slapd-<*hostname*>

3 Create a self-signed certificate

/opt/SUNWdsee/ds6/bin/dsadm add-selfsign-cert -S "cn=<*machine name with domain*>,O=<*preferred root suffix*>"/<*DS-server-root*>/slapd-<*hostname*>/<*certificate name*>

Where S = Create an individual certificate and add it to database,the second variable represents the path of Directory Server instance and the last variable is for the certificate alias.

4 Set the server properties to this certificate

/opt/SUNWdsee/ds6/bin/dsconf set-server-prop -p *non-ldap-port*
ssl-rsa-cert-name:<*certificate name*>

5 Restart the DS

/opt/SUNWdsee/ds6/bin/dsadm restart /<*DS-server-root*>/slapd-<*hostname*>/

6 Now stop the DS and remove the default Cert (this ensures that the above generated certificate will be the default cert)

/opt/SUNWdsee/ds6/bin/dsadm stop /<*DS-server-root*>/slapd-<*hostname*>/

7 Now remove the default certificate

/opt/SUNWdsee/ds6/bin/dsadm remove-cert /<*DS-server-root*>/slapd-<*hostname*>/
defaultCert

where the first variable represents the slapd-path and the second variable represents the alias of the certificate. In case you want to export the above default cert, following is the command

```
/opt/SUNWdsee/ds6/bin/dsadm export-cert -o /<any path>/slapd-cert.export
/<DS-server-root>/slapd-<hostname>/ <original default cert alias>
```

where o=output file (/<any path>/slapd-cert.export),the second variable represents the slapd-path and the third variable represents the certificate alias.

Retrieving the CA Certificate from the Directory Server Certificate Database

Ensure that you have enabled SSL in Directory Server. To export the Directory Server certificate to a temporary file so that you can import it into the certificate database of the Directory Server Connector, issue the following command:

```
<ISW-server-root>\shared\bin\certutil.exe -L -d .
-P slapd-hostname- -n server-cert -a \ > C:\s-cert.txt
```

ISW-server-root is the path where ISW-hostname directory is present.

These examples are run in the `alias` directory immediately below the server root. Otherwise, Directory Server will not find the certificate database.

Retrieving the CA Certificate from the Directory Server (using dsadm command on Solaris platform)

Ensure that you have enabled SSL in Directory Server. To retrieve the CA certificate issue the following command:

```
/opt/SUNWdsee/ds6/bin/dsadm export-cert -o /<any path>
/slapd-cert.export /<DS-server-root>/slapd-<hostname>/
<original default cert alias>
```

Enabling SSL in the Active Directory Connector

Identity Synchronization for Windows *automatically* retrieves Active Directory SSL certificates over SSL and imports them into the Connector's certificate database using the same credentials you provided for the Connector.

However; if an error occurs (for example, invalid credentials or no SSL certificates were found), you can retrieve an Active Directory CA certificate and add it to the Connector certificate database. See the following sections for instructions:

- "Retrieving an Active Directory Certificate" on page 244
- "Adding Active Directory Certificates to the Connector's Certificate Database" on page 245

Retrieving an Active Directory Certificate

If an error occurs, you can use certutil (a program that ships with Windows 2000/2003) or LDAP to retrieve an Active Directory certificate, as described in the following sections.

Note – The certutil command discussed in this section is *not* the same as the certutil command that ships with the Directory Server and discussed previously in this publication.

Using Window's Certutil

▼ To Retrieve an Active DirectoryCertificate Using the certutil **program**

1 Run the following command from the Active Directory machine to export the certificate.

```
C:\>certutil -ca.cert cacert.bin
```

2 You can then import thecacert.bin **file into a certificate database.**

Using LDAP

▼ To Retrieve an Active Directory Certificate using LDAP

1 Execute the following search against Active Directory:

```
ldapsearch -h CR-hostname -D administrator_DN -w administrator_password
-b "cn=configuration,dc=put,dc=your,dc=domain,dc=here" "cacertificate=*"
```

Where the *administrator_DN* might look like:

```
cn=administrator,cn=users,dc=put,dc=your,dc=domain,dc=here
```

In this example, the domain name is: *put.your.domain.name.here.*

Several entries will match the search filter. You probably need the entry using cn=Certification Authorities, cn=Public Key Services in its DN.

2 Open a text editor and cut the first value of the first CA certificate attribute (it should be a base64 encoded text block). Paste that value (text block) into the text editor (only the value). Edit the contents, so that none of the lines start with white space.

3 Add `-----BEGIN CERTIFICATE-----` **before the first line and** `-----END CERTIFICATE-----` **after the last line. See the following example:**

```
-----BEGIN CERTIFICATE-----
MIIDvjCCA2igAwIBAgIQDgoyk+Tu14NGoQnxhmNHLjANBgk
qhkiG9w0BAQUFADCBjjEeMBwGCSqGSIb3DQEJARYPYmVydG
9sZEBzdW4uY29tMQswCQYDVQQGEwJVUzELMAkGA1UECBMCV
FgxDzANBgNVBAcTBkF1c3RpbjEZMBcGA1UEChMQU3VuIE1p
Y3Jvc3lzdGVtczEQMA4GA1UECxMHaVBsYW5ldDEUMBIGA1U
EAxMLUmVzdGF1cmFudHMwHhcNMDIwMTExMDA1NDA5WhcNMT
IwMTExMDA1OTQ2WjCBjjEeMBwGCSqGSIb3DQEJARYPYmVyd
G9sZEBzdW4uY29tMQswCQYDVQQGEwJVUELMAkGA1UECBMCV
FgxDzANBgNVBAcTBkF1c3RpbjEZMBcGA1UEChMQU3VuIE1p
Y3Jvc3lzdGVtczEQMA4GA1UECxMHaVBsYW5ldDEUMBIGA1U
EAxMLUmVzdGF1cmFudHMwXDANBgkqhkiG9w0BAQEFAANLAD
BIAkEAyekZa8gwwhw3rLK3eV/12St1DVUsg31LOu3CnB8cM
HQZXlgiUgtQ0hm2kpZ4nEhwCAHhFLD3iIhIP4BGWQFjcwID
AQABo4IBnjCCAZowEwYJKwYBBAGCNxQCBAYeBABDAEEwCwY
DVR0PBAQDAgFGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBB
YEFJ5Bgt6Oypq7T8Oykw4LH6ws2d/IMIIBMgYDVR0fBIIBK
TCCASUwgdOggdCggc2GgcpsZGFwOi8vL0NOPVJlc3RhdXJh
bnRzLENOPWRvRvd2l0Y2hlcixDTj1DRFAsQ049UHVibGljJTI
wS2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZm
lndXJhdGlvbixEQz1yZXN0YXVyYW50cyxEQz1jZW50cmFsL
RPXN1bixEQz1jb20/Y2VydGlmaWNhdGVSZXZvY2F0aW9u
TGlzdD9iYXNlP29iamVjdGNsYXNzPWNSTERpc3RyaWJ1dGl
vblBvaW50ME2gS6BJhkdodHRwOi8vZG93aXRjaGVyLnJlc3
RhdXJhbnRzLmNlbnRyYWwuc3VuLmNvbS9DZXJ0RW5yb2xsL
1Jlc3RhdXJhbnRzLmNybDAQBgkrBgEEAYI3FQEEAwIBADAN
BgkqhkiG9w0BAQUFAANBAL5R9R+ONDdVHWu/5Sd9Tn9dpxN
8oegjS88ztv1HD6XSTDzGTuaaVebSZV3I+ghSInsgQbH0gW
4fGRwaI BvePI4=
-----END CERTIFICATE-----
```

4 Save the certificate into a file (such as `ad-cert.txt`).

5 You can then import that file (for example, `ad-cert.txt`) into a certificate database. Continue to the next section,

Adding Active Directory Certificates to the Connector's Certificate Database

Use this procedure only if you enabled SSL for the Active Directory Connector after installing the Connector or if invalid credentials were provided during installation.

▼ **To Add Active Directory Certificate to the Connector's Certificate Database**

1   **On the machine where the Active Directory Connector is installed, stop the Identity Synchronization for Windows service/daemon.**

2   **Retrieve the Active Directory CA certificate using one of the following methods:**
    -
    -

3   **Assuming the Active Directory Connector has connector ID CNN101 (see** `logs/central/error.log` **for a mapping from connector ID to the directory source it manages), go to its certificate database directory on the machine where it was installed, and import the certificate file:**
    - If the certificate was retrieved using `certutil`, type:

        `<`*ISW-server-root*`>\shared\bin\certutil.exe -A -d . -n ad-ca-cert -t C,, -i \cacert.bin`
    - If the certificate was retrieved using LDAP, type:

        `<`*ISW-server-root*`>\shared\bin\certutil.exe -A -d . -n ad-ca-cert -t C,,`
        `-a -i \ad-cert.txt`

        ISW-server-root is the path where ISW-hostname directory is present

        The certificate can be imported using dsadm in the following manner(on Solaris platform):

`/opt/SUNWdsee/ds6/bin/dsadm add-cert -C <`*DS-server-root*`>/slapd-<`*hostname*`>/ ad-ca-cert cacert.bin`

        where ad-ca-cert is the name of the certificate assigned after the import and cacert.bin is the certificate about to be imported

4   **Restart the Identity Synchronization for Windows service/daemon.**

> **Note –** Because the Directory Server `certutil.exe` is installed automatically when you install Directory Server 6.0, you will not be able to add a CA certificate to a connector installed on a machine with no Directory Server.
>
> At a minimum, you must install the Sun Java System Server Basic Libraries and Sun Java System Server Basic System Libraries from the Directory Server 6.0 package on the server where the Active Directory Connector is installed. (You do not have to install the Administration Server or Directory Server components.)
>
> In addition, be sure to select the JRE subcomponent from the Console (to ensure your ability to uninstall).

Adding Active Directory Certificates to Directory Server

> **Note –** Make sure that you have enabled SSL in Directory Server.

▼ To Add the Active Directory CA certificate to the Directory Server Certificate Database

1. **Retrieve the Active Directory CA certificate using one of the following methods:**
   - "Using Window's Certutil" on page 244
   - "Using LDAP" on page 244

2. **Stop Directory Server.**

3. **Import cacert.bin into the** `<DS-server-root>\slapd-hostname\alias` **folder on Windows and for Solaris and Linux import it into** `<DS-server-root>/slapd-hostname/alias` **directory.**

4. **On the machine where Directory Server is installed, import the Active Directory CA certificate as follows:**
   - If the certificate was retrieved using `certutil`, type:

     ```
     <ISW_server_root>\shared\bin\certutil.exe -A -d .
     -P slapd-hostname- -n ad-ca-cert -t C,, -i \cacert.bin
     ```
   - If the certificate was retrieved using LDAP, type:

     ```
     <ISW_server_root>\shared\bin\certutil.exe -A -d .
     -P slapd-hostname- -n ad-ca-cert -t C,, -a -i \ad-cert.txt
     ```

ISW-server-root is the path where ISW-hostname directory is present

- If the certificate was retrieved using dsadm (on Solaris), type:

```
/opt/SUNWdsee/ds6/bin/dsadm add-cert -C <DS-server-root>
/slapd-<hostname>/ ad-ca-cert cacert.bin
```

Where ad-ca-cert is the name of the certificate assigned after the import and cacert.bin is the certificate about to be imported

5 Start Directory Server.

Adding Directory Server Certificates to the Directory Server Connector

If you enable SSL communication between the Directory Server Plug-in and Active Directory, then you must add the Active Directory CA Certificate to the certificate database of each Directory Server master.

▼ To Add the Directory Server Certificates to the Directory Server Connector

1 On the machine where the Directory Server Connector is installed, stop the Identity Synchronization for Windows service/daemon.

2 Retrieve the Directory Server CA certificate.

3 Assuming the Directory Server Connector has connector ID CNN100 (see `logs/example/error.log` **for a mapping from connector ID to the directory source it manages), go to its certificate database directory on the machine where it was installed, and import the** `cacert.bin` **file:**

```
<ISW_server_root>\shared\bin\certutil.exe -A -d . -n ds-cert -t C,, -i C:\s-cert
```

ISW-server-root is the path where ISW-hostname directory is present.

4 Restart the Identity Synchronization for Windows service/daemon.

11

Understanding Audit and Error Files

Identity Synchronization for Windows provides information about the installation and configuration status, the day-to-day system operations, and any error conditions that are related to your deployment.

This chapter explains how to access and understand this information in the following sections:

Understanding the Logs

You can view various types of information from the Status tab of the Identity Synchronization for Windows Console.

If you select one of the following nodes in the navigation tree pane (on the left), the content presented on the Status tab changes to provide specific information about that item.

- **Directory Source**: Select a directory source node (such as dc=example,dc=com) to view status information about that directory source.

- **To Do**: Select this node for a list of the steps you must complete to successfully install and configure Identity Synchronization for Windows (the program greys-out all completed steps).

- **Audit File**: Select this node for information about day-to-day system operations (including error conditions).

- **Error File**: Select this node for information about error conditions on your system. (The Error log essentially acts as a filter in which only the error entries are displayed.)
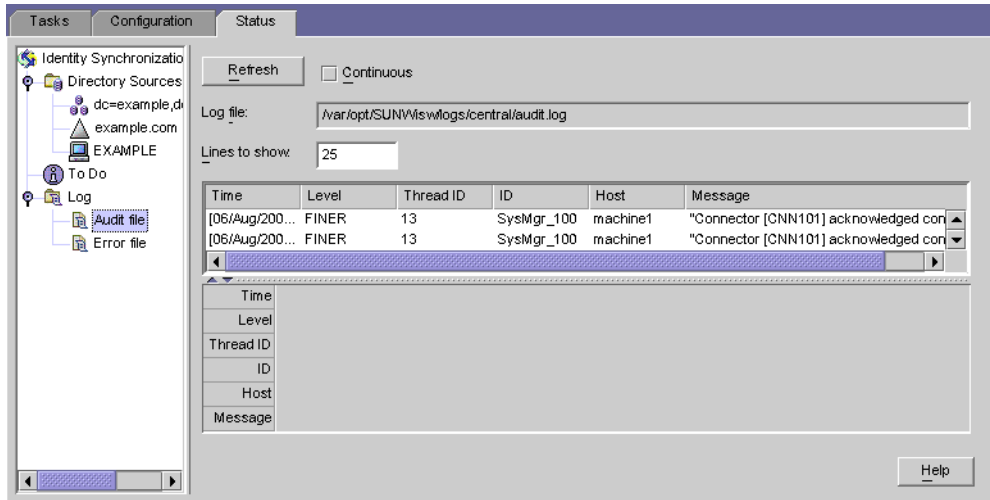
Log Types

This section describes the different kinds of logs that are available for Identity Synchronization for Windows:

- "Central Logs" on page 250
- "Local Component Logs" on page 251
- "Local Windows NT Subcomponent Logs" on page 252
- "Directory Server Plug-in Logs" on page 252

Central Logs

As long as Identity Synchronization for Windows components can access Message Queue, all audit and error messages will be logged in the Identity Synchronization for Windows central logger. Consequently, these central logs (which include messages from all components) are the primary logs to monitor.

The centralized logs are located on the machine where Core is installed, in the following directories:

- **On Solaris**: /var/opt/SUNWisw/logs
- **On Linux**: /var/opt/sun/isw/logs
- **On Windows**: *installation_root*/isw-*machine_name*/logs/central/

TABLE 11–1   Log Types for Identity Synchronization for Windows

Log Name	Description

TABLE 11–1 Log Types for Identity Synchronization for Windows   *(Continued)*

error.log	Warning and Severe messages are reported here.
audit.log	A superset of error.log that includes messages about each synchronization event.
resync.log	Messages generated by the resync command are reported here.

Each central log also includes information about each component ID. For example,

```
[2003/03/14 14:48:23.296 -0600] INFO 13
"System Component Information:
SysMgr_100 is the system manager (CORE);
console is the Product Console User Interface;
CNN100 is the connector that manages
[example.com (ldaps:// server1.example.com:636)];
CNN101 is the connector that manages
[dc=example,dc=com (ldap:// server2.example.com:389)];"
```

In addition to the central logger, each component has it's own local logs. You can use these local logs to diagnose problems with the connector if it cannot log to the central logger.

Local Component Logs

Each connector, the system manager, and the central logger have the following local logs:

TABLE 11–2 Local Logs

Log Name	Description
audit.log	A superset of error.log that includes messages about each synchronization event. These messages are also written to the central audit.log.
error.log	Warning and Severe messages are reported here. These messages are also written to the central error.log.

These local logs are located in the following subdirectories:

- **On Solaris**: /var/opt/SUNWisw/logs
- **On Linux**: /var/opt/sun/isw/logs
- **On Windows**: *installation_root*/isw-*machine_name*/logs/central/

  The sysmgr and clogger100 (central logger) directories are on the machine where Core is installed.

  Identity Synchronization for Windows rotates these local component logs daily by moving the current log to a log file that includes the date, as follows:

  ```
  audit_2004_08_06.log
  ```

> **Note –** By default, Identity Synchronization for Windows deletes connector logs after ten days. You can extend this period by editing the
> `com.sun.directory.wps.logging.maxmiumDaysToKeepOldLogs` value in the `Log.properties` file and restarting the service daemon.

Local Windows NT Subcomponent Logs

The following Windows NT subcomponents also have local logs:

- Windows NT Change Detector DLL
- Password Filter DLL

  These subcomponent logs are located in the `SUBC1XX` (for example, `SUBC100`) subdirectories of the following directory:

  installation_root/isw-*machine_name*/`logs`/

  Identity Synchronization for Windows limits these files to 1 MB in size, and keeps only the last 10 logs.

Directory Server Plug-in Logs

The Directory Server Plug-in logs information through the Directory Server connector to the central log and through the Directory Server logging facility. Consequently, local Directory Server Plug-in log messages will also be saved in the Directory Server error log.

Directory Server saves information into the error log from other Directory Server Plug-ins and components. To identify messages from the Identity Synchronization for Windows Directory Server Plug-in, you can filter out lines containing the `isw` string.

By default, only minimal Plug-in log messages are displayed in the error log. For example:

```
[14/Jun/2004:17:08:36 -0500] - ERROR<38747> - isw - conn=-1
op=-1 msgId=-1 - Plug-ins unable to establish connection to DS Connector
at attila:1388, will retry later
```

▼ To Change the Verbosity Level of the Error Logs

You can change the default verbosity level of the Directory Server error log through DSCC as follows:

1 Log in to Directory Service Control Center.

2 On the Directory Servers tab page, click the server whose log level you want to configure.

3 Select the Server Configuration tab, then the Error Logging tab.

4   **In the General → Additional Items to Log section, select Plug-Ins.**

5   **Click Save.**

You can enable plug-in logging using the command line.

```
$ dsconf set-log-prop errors level:err-plugins
```

For more information about Directory Server logging, refer to Chapter 14, "Directory Server Logging," in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide* in *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

Reading the Logs

Every log message includes the following information:

- **Time:** Indicates when (time and date) the log entry was generated. For example:

  ```
  [13/Aug/2004:06:14:36:753 -0500]
  ```

- **Level:** Indicates the severity and verbosity of the log message. Identity Synchronization for Windows uses the following log levels:

TABLE 11–3   Log Levels

Log Level	Description
INFO	These messages provide a minimum amount of information about each action so you can see that the system is running correctly. For example, you can see when a change is detected and when synchronization occurs. These messages are always logged to the audit log.
FINE	These messages contain more information about an action as it travels through the system.
FINER	These messages contain even more information about an action as it travels through the system. Turning the logging level to FINER for all components may impact performance.
FINEST	These messages contain the most information about an action as it travels through the system. Turning the logging level to FINEST for all components may significantly impact performance.

- **Thread ID:** Displays the Java thread ID of the function causing the event.
- **ID:** Identifies the component (console, system manager, and so forth.) causing the event.
- **Host:** Displays the name of the host causing the event.
- **Message:** Displays audit or error information associated with the event. Some examples include:

```
"Resetting Central Logger configuration ..."
"System manager is shutting down."
"Processing request (ID=ID_number
 from the console to stop synchronization."
```

Configuring Your Log Files

▼ To Configure Logging for Your Deployment

1   Open the Console and select the Configuration tab.

2   In the navigation tree pane, and expand nodes until you see the Logs node.

3   Select the Logs node and the Log Files panel is displayed on the Configuration tab.
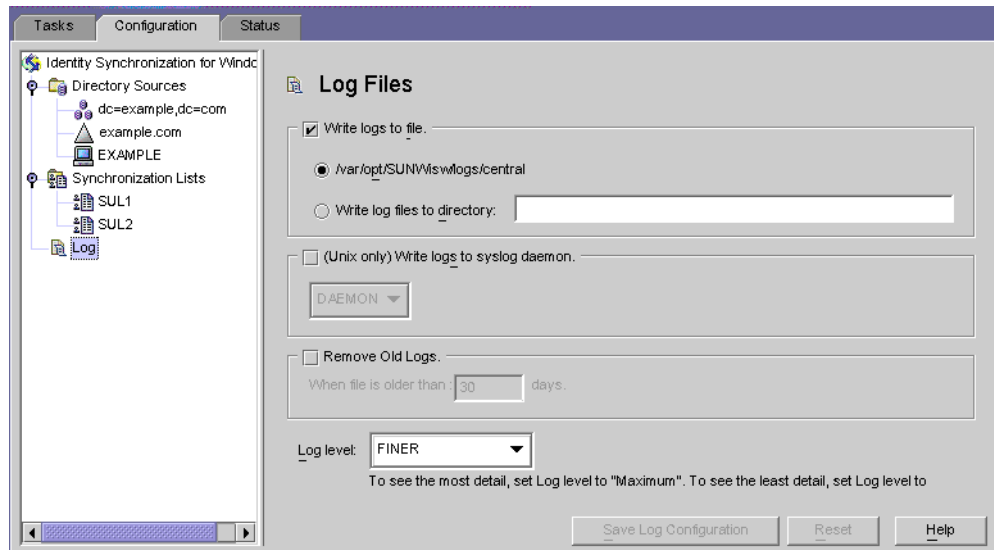
4   Use the Log Files pane to configure your log files, as follows:

- **Write logs to file.** Enable this option to write logs to a file on the Core host.

  After selecting this option you can:

  - Enable the default log directory and file (for example,
    /var/opt/SUNWisw/logs/central).

  - Enable the Write log files to directory option, and then specify a path and file name for
    the log file.

> **Note –** The Console does not verify whether a specified log file location actually exists. The central logger will try to create the log directory if it does not exist. Consequently, there is no indication that you specified and saved a nonexistent log location until you try to view the logs. After several attempts to view the logs, a message displays to report the Console's inability to find logs at the specified location.

- **On Solaris Only** — *Write logs to syslog daemon*: Enable this option if Identity Synchronization for Windows resides on a Solaris platform. Use the drop-down list to select a category for writing the log. (*Default is DAEMON.*)

> **Note –** When you select this option, Identity Synchronization for Windows logs everything to the syslog; however, the syslog is configured by default to log WARNING and SEVERE messages only.
>
> To configure syslog to log INFO messages, edit /etc/syslog.conf and change the following line:
>
> `*.err;kern.debug;daemon.notice;mail.crit /var/adm/messages`
>
> to
>
> `*.err;kern.debug;daemon.notice;daemon.info;mail.crit /var/adm/messages`
>
> After making this change, you must restart the syslog daemon as follows:
>
> `/etc/init.d/syslog stop ; /etc/init.d/syslog start`
>
> To enable FINE, FINER, and FINEST logging, include daemon.debug in the semi-colon separated list.

- **Remove Old Logs**: The number of log files will continue to grow (one per day) indefinitely. To avoid running out of disk space, enable this option and specify when the program can delete old logs from the central log file.

  For example, if you specify 30 days, Identity Synchronization for Windows will delete all files when they become 31 days old.

- **Log Level**. Use the drop-down list to select the level of detail you want to see in your system logs. (Review "Reading the Logs" on page 253.)

5 Click the Save Log Configuration button to create log files based on the selected options.

Viewing Directory Source Status

▼ To View the Status of your Directory Sources

1   **From the Identity Synchronization for Windows Console, select the Status tab.**

2   **In the navigation tree pane, expand the Directory Source node, and then select the directory source node (such as dc=example,dc=com).**

The Status tab content changes to provide information related to the selected directory source.

Note – When viewing the Directory Source status you are essentially viewing the status of the connector associated with that Directory Source.

Click Update to refresh the information on this tab. The following information is provided on the Status tab:

- **State:** Reflects the current state of the directory source. Valid states include:
    - **Uninstalled:** The connector has not been installed.
    - **Installed:** The connector is installed, but is not ready for synchronization because it has not received its runtime configuration yet. If the connector remains in this state for more than a minute, something is probably wrong.
    - **Ready:** The connector is ready for synchronization, but it is currently not synchronizing any objects. A connector remains in the Ready state if synchronization has not been started or if synchronization has been started but not all subcomponents have established connections with the connectors.
    - **Syncing:** The connector is synchronizing objects. There might still be errors, so consult the error log if you notice that changes are not synchronized.
- **Active**: Indicates whether the directory source is active or down.
- **Last Communication:** Indicates the time of the last response from this directory source's connector.

Viewing Installation and Configuration Status

▼ To View the Remaining Steps of the Installation and Configuration Process

1   From the Identity Synchronization for Windows Console, select the Status tab.

2   In the navigation tree pane, expand the To Do node.

The Status tab content changes to provide a checklist of the installation and configuration steps (for example, see "Viewing Directory Source Status" on page 256).
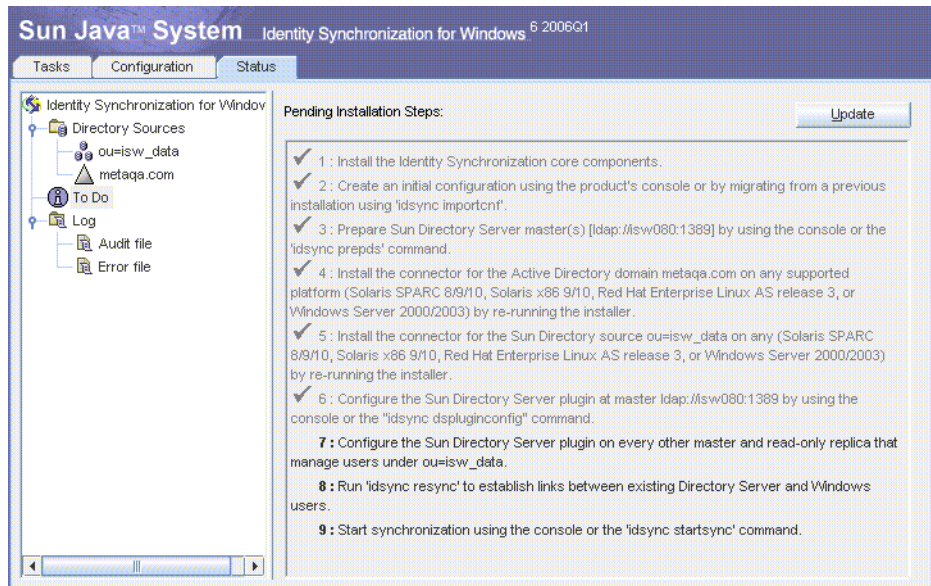
3   Click the Update button (upper right) to refresh the list.

Completed steps will be check-marked and greyed-out. You must complete the remaining steps to successfully complete the installation and configuration process.

Viewing Audit and Error Logs

▼ To View Your Error Logs

1 **From the Identity Synchronization for Windows Console, select the Status tab.**

2 **In the navigation tree pane, expand the Audit File or the Error File node.**

The Status tab content changes to display the current logs.
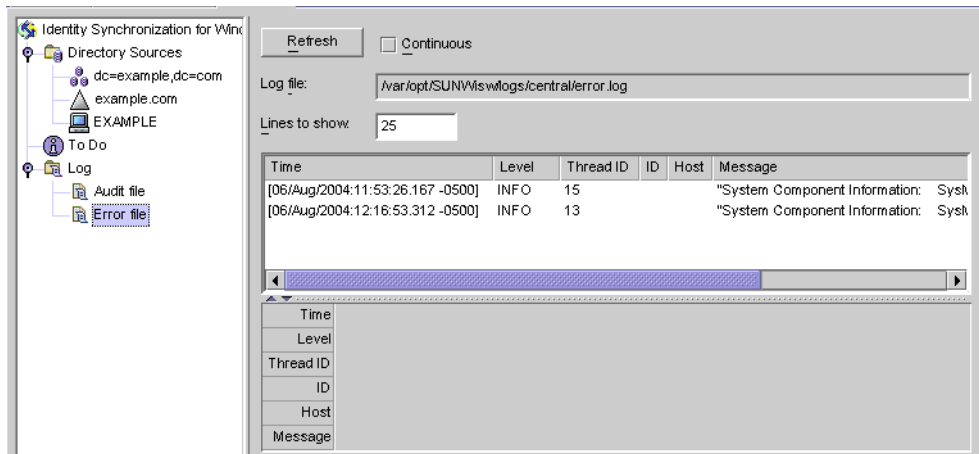
Click Refresh to load the latest audit or error information.

The following information is provided on the Status tab:

- **Continuous:** Updates and displays the latest audit or error information constantly.

- **Log File:** Displays the full path name of the audit or error log being read; for example:

  C:\Program Files\Sun\MPS\isw-*hostname*\logs\central\audit.log

- **Lines to show:** Specifies how many audit or error entries to display. (*Default is 25.*)

Enabling Auditing on a Windows NT Machine

If you have a Windows NT machine in your deployment, verify that auditing is enabled or Identity Synchronization for Windows cannot log messages from that machine.

▼ To Enable Audit Logging on Your Windows NT Machine

1   **From the Windows NT Start menu, select Programs > Administrative Tools > User Manager for Domains.**

2   **When the User Manager dialog box is displayed, select Policies > Audit from the menu bar.**
    The Audit Policy dialog box is displayed.

3   **Enable the Audit These Events button and then enable the Success and Failure boxes.**

4   **Click OK to close the dialog box.**
    These settings will remain in effect until you change them again.

12

Troubleshooting

This chapter provides information to help you troubleshoot problems you may encounter while using Identity Synchronization for Windows. The information is organized as follows:

Before you start the troubleshooting process, make sure you have installed the correct versions of the following:

- Operating System versions and patches/service packs
- Directory Server versions and patches

For information on Operating System and Directory Server correct versions and patches, see *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*.

Troubleshooting Checklist

> **Note –** Administrators: When you are debugging problems, adjust the logging level (as described in "Configuring Your Log Files" on page 254 problems.

Some events (such as the program failing to synchronize a user change because the user was not included in the SUL) are not included in a log file until you adjust the log level to FINE or higher. The log level should be left at INFO during all idsync resync operations.

The `idsync printstat` command can be a useful tool as you install and configure Identity Synchronization for Windows. When you run `printstat` (see "Using printstat" on page 296), it displays a list of the remaining steps you have to perform to complete the installation and configuration process.

▼ To Troubleshoot Issues with Identity Synchronization for Windows 6.0

1 Are there any problems reported in the central error.log?

isw-*hostname*/logs/central/error.log

Almost all errors will be reported in the central error log file. Also, additional information about any error is usually available in the audit.log file. To ease correlation of related log entries, the audit.log file also includes all entries in the error log.

2 Is this problem explained as a known issue in the Release Notes document?

3 Was the installation performed on a clean machine? Problems might occur when this product is reinstalled if the uninstallation of the previous configuration was not complete. Please refer to Chapter 9.

4 Was the Core properly installed? If Core installation completed successfully, then log files will exist in the isw-*hostname*/logs/central/ **directory.**

5 Was the Directory Server running during resource configuration?

6 Is the Core, including the Message Queue and the System Manager, currently running? On Windows, check for the appropriate service name. On Solaris and Linux, check for the appropriate daemon name. Use the idsync printstat **command to verify that the Message Queue and System Manager are active.**

7 Was a configuration saved successfully? If the idsync printstat command lists connectors, then a configuration was saved successfully.

8   **Were all connectors installed? One connector must be installed for each directory source being synchronized.**

9   **Were all subcomponents installed? Directory Server and Windows NT Connectors require subcomponents to be installed after the Connector installation. The Directory Server Plug-in must be installed in each Directory Server replica.**

10   **Were post-installation procedures followed? The Directory Server must be restarted after the Directory Server Plug-in is installed. The Windows NT Primary Domain Controller must be restarted after the Windows NT subcomponents are installed.**

11   **Was synchronization started either from the Console or command line?**

12   **Are all connectors currently running?**

13   **Verify that all connectors are in the SYNCING state using the Console or idsync printstat.**

14   **Are the directory sources being synchronized currently running?**

15   **Verify using the Console that modifications and/or creates are synchronized in the expected direction(s).**

16   **If synchronizing users and groups that existed in only one directory source, were these users and groups created in the other directory source using the** `idsync resync` **command?**

Note – You must run `idsync resync` whenever there are existing users and groups. If you do not resynchronize existing users, resynchronization behavior remains undefined.

---

17   **If synchronizing users that existed in both directory sources, were these users linked using the** `idsync resync` **command?**

18   **If user creates fail from Active Directory or Windows NT to the Sun Java System Directory Server, verify that all mandatory attributes in the Directory Server objectclass are specified as creation attributes and values for the corresponding attributes are present in the original user entry.**

19   **If synchronizing creates from Directory Server to Windows NT and the user creation succeeded, but the account is unusable, verify that the user name does not violate Windows NT requirements.**

For example, if you specify a name that exceeds the maximum allowable length for Windows NT, the user will be created on NT but will remain unusable and uneditable until you rename the user (User → Rename).

20 **For the Windows NT SAM Change Detector subcomponent to be effective, you must turn on the NT audit log. Select Start → Programs → Administrative Tools → User Manager, and then select Policies → Audit Policies. Select Audit These Events and then both the Success and Failure boxes for User and Group Management.**

Select Event Log Settings in the Event Viewer → Event Log Wrapping, and then select Overwrite Events as Needed.

21 **Are the users that fail to synchronize within a Synchronization User List? For example, do they match the base DN and filter of a Synchronization User List? In deployments that include Active Directory, on-demand password synchronization fails silently if the Sun Java System Directory Server entry is not in any Synchronization User List. This most often occurs because the filter on the Synchronization User List is incorrect.**

22 **Were the synchronization settings changed? If the synchronization settings changed from only synchronizing users from Active Directory to the Sun Java System Directory Server to synchronizing users from the Directory Server to Active Directory, then the Active Directory SSL CA certificate must be added to the connector's certificate database. The** `idsync certinfo` **command reports what SSL certificates must been installed based on the current SSL settings.**

23 **Are all host names properly specified and resolvable in DNS? The Active Directory domain controller should be DNS-resolvable from the machine where the Active Directory Connector is running and the machine where the Sun Java System Directory Server Plug-in is running.**

24 **Does the IP address of the Active Directory domain controller resolve to the same name that the connector uses to connect to it?**

25 **Does the source connector detect the change to the user? Use the central** `audit.log` **to determine if the connector for the directory source where the user was added or modified detects the modification.**

26 **Does the destination connector process this modification?**

27 **Are multiple Synchronization User Lists configured? If so, are these in conflict? More specific Synchronization User Lists should be ordered before less specific ones using the Console.**

28 **If flow is set to bidirectional or from Sun to Windows and there are Active Directory data sources in your deployment, are the connectors configured to use SSL communication?**

29 **If memory problems are suspected on Solaris or Linux environments check the processes. To view which components are running as different processes, enter**

```
/usr/ucb/ps -gauxwww | grep com.sun.directory.wps
```

The output gives the full details including the ID of connectors, system manager and central logger. This can be useful to see if any of the processes are consuming excessive memory.

30 **If you are creating or editing the Sun Java System Directory source, and the Directory Server does not display in the Choose a known server drop-down list, check that the Directory Server is running. The Directory Server must be running to appear in the drop down list of available hosts.**

If the server in question is down temporarily, type the host and port into the Specify a server by providing a hostname and port field.

Note – Identity Synchronization for Windows uses a short host name by default; however, the default host name may not work with your configuration. We recommend using a fully qualified name whenever you are asked to provide a host name.

---

31 **Do you receive the following error while running uninstaller program?**

```
./runInstaller.sh
IOException while making /tmp/SolarisNativeToolkit_5.5.1_1
executable:java.io.IOException: Not enough space java.io.IOException: Not enough space
```

Increase the size of the swap file mounted at /tmp.

Troubleshooting Connectors

Use the information in this section to troubleshoot problems with your connectors. The information is organized as follows:

- "How to Determine the ID of a Connector Managing a Directory Source?" on page 265
- "How to Determine a Connector's Current State?" on page 266

How to Determine the ID of a Connector Managing a Directory Source?

You can use one of the following methods to determine the connector ID:

- "Using the Central Logs" on page 265
- "Using idsync printstat" on page 266

Using the Central Logs

Determine the connector IDs of the directory sources being synchronized by looking in the central audit.log. At startup, the central logger logs the IDs of each connector and the directory source that it manages. Look for the last instance of the startup banner for the most recent information.

For example, in the following log message there are two connectors:

- **CNN101** is a Sun Directory Connector that manages dc=example,dc=com

- **CNN100** is an Active Directory Connector that manages the example.com domain

```
[2006/03/19 00:00:00.722 -0600] INFO    16
"System Component Information:
SysMgr_100 is the system manager (CORE);
console is the Product Console User Interface;
CNN101 is the connector that manages
[dc=example,dc=com (ldap://host1.example.com:389)];
CNN100 is the connector that manages
[example.com (ldaps://host2.example.com:636)];"
```

Using idsync printstat

The connector IDs and status are also available from the idsync printstat command (see).

A sample output of this command follows:

```
Connector ID: CNN100
Type:    Active Directory
Manages:  example.com (ldaps://host2.example.com:636)
State:    READY
Connector ID: CNN101
Type:    Sun Java System Directory
Manages: dc=example,dc=com
(ldap://host1.example.com:389)
State:    READY
Sun Java System
Message Queue Status:  Started
Checking the System Manager status over the Sun Java System
Message Queue.
System Manager Status:  Started SUCCESS
```

How to Determine a Connector's Current State?

You can determine the current state of the connectors involved in synchronization, using the Status pane in the Console, the idsync printstat command (as shown previously), or by looking in the central audit.log.

Search for the last message in the audit.log that reports the connector state. For example, in the following log message you can see that connector CNN101 is in the READY state.

```
[2006/03/19 10:20:16.889 -0600]
 INFO    13 SysMgr_100 host1
  "Connector [CNN101] is now in state "READY"."
```

How to Determine a Connector's Current State describes the different connector states.

TABLE 12–1 Connector State Meanings

State	Meaning
UNINSTALLED	The connector has not been installed.
INSTALLED	The connector has been installed, but it has not received its configuration.
READY	The connector has been installed and has received its configuration, but it has not started to synchronize.
SYNCING	The connector has been installed, has received its configuration, and has attempted to start synchronizing.

What to Do if the Connector is in the UNINSTALLED State?

Install the connector.

What to Do if the Connector Install Failed but You Cannot Reinstall?

If the connector installation failed, but the Identity Synchronization for Windows installation program thinks that the connector is installed, the installation program will not allow you to reinstall the connector.

Run idsync resetconn (as described in "Using resetconn" on page 297) to reset the connector's state to UNINSTALLED, and then re-install the connector.

What to Do if the Connector is in the INSTALLED State?

If a connector remains in the installed state for a long period of time, then most likely it is not running, or it is unable to communicate with the Message Queue.

At the machine where the connector was installed, look in the connector's logs (audit.log and error.log) for potential errors. If the connector cannot connect to the Message Queue, then that error will be reported here. If this is the case, see "Troubleshooting Message Queue" on page 274 for possible causes.

If the most recent messages in the audit log are old, then perhaps the connector is not running. See "Troubleshooting Components" on page 268.

What to Do if the Connector is in the READY State?

A connector remains in the READY state until synchronization has been started and all of its subcomponents have been installed and have connected to the connector. If synchronization has not been started, then start it using the Console or command line utility.

If synchronization has been started, but a connector does not enter the SYNCING state, then there is likely a problem with subcomponent. See "Troubleshooting Subcomponents" on page 272

What to Do if the Connector is in the SYNCING State?

If all connectors are in the SYNCING state, but modifications are not being synchronized, then verify that the synchronization settings are correct:

- Using the Console, verify that modifications and/or creates are synchronized in the expected direction (for example, from Windows to the Sun Java System Directory Server).

- Using the Console, verify that the attribute being modified is a synchronized attribute (note: passwords are always synchronized). If created user entries are not being synchronized, then verify that user creation flow is enabled in the Console.

- Does the source connector detect the change to the user? Use the central `audit.log` to determine if the connector for the directory source where the user was added or modified detects the modification. Does the destination connector process this modification?

What to Do if the Active Directory Connector Fails to Contact Active Directory Over SSL?

If the Active Directory Connector fails to contact Active Directory over SSL and the following error message displays, restart the AD domain controller.

```
Failed to open connection to
ldaps://server.example.com:636,
error(91): Cannot connect to the LDAP server,
reason: SSL_ForceHandshake failed: (-5938)
Encountered end of file.
```

What to Do if Detecting and Applying Changes in Active Directory Fails?

If a non-admin account is used for the Active Directory connector, then the default permissions for this user are not sufficient. Some operations such as a resync process from Active Directory to Directory Server succeeds, but other operations such as detecting and applying changes in Active Directory could fail abruptly. For example, if you synchronize the deletions from Active Directory to Directory Server, then even full control is insufficient. To resolve this, you must use a Domain Administrator account for the Active Directory connector.

Troubleshooting Components

Use the information in this section to troubleshoot components. The information is organized as follows:

- "Examining WatchList.properties" on page 271

On Solaris

The command /usr/ucb/ps -auxww | grep com.sun.directory.wps will list all of the Identity Synchronization for Windows processes running. This table shows which processes should be running.

TABLE 12–2   Identity Synchronization for Windows Processes

Java Process Class Name	Component	When Present
com.sun.directory.wps.watchdog.server.WatchDog	System Watchdog	Always
com.sun.directory.wps.centrallogger.CentralLoggerManager	Central Logger	Only where Core is installed
com.sun.directory.wps.manager.SystemManager	System Manager	Only where Core is installed
com.sun.directory.wps.controller.AgentHarness	Connector	One for each connector installed

If the expected number of processes are not running, then issue the following commands to restart all Identity Synchronization for Windows processes.

```
# /etc/init.d/isw stop
# /etc/init.d/isw start
```

If the WatchDog process is running, but the expected number of java.exe processes are not running, then see the "Examining WatchList.properties" section to verify that all components were installed properly.

Like other system components, the Sun Java System Directory Server Plug-in sends log records over the bus that are managed by the central logger for end-user viewing. However, the Plug-in also logs some messages that may not show up over the bus (for instance when the subcomponent cannot contact the connector). In this case the log messages only show up in the Plug-in's logs directory on the file system, which should look something like the following:

serverroot/isw-*hostname*/logs/SUBC*id*

Because the Plug-in runs with the Directory Server process, there could potentially be a problem for the Plug-in's ability to write into its logs directory. This happens if the directory server runs as a different user than the owner of the logs directory. In this case, it may be necessary to give the Plug-in permission explicitly by changing the directories permission or owner using native operating system commands.

On Linux

The command /usr/ucb/ps -auxww | grep com.sun.directory.wps will list all of the
Identity Synchronization for Windows processes running. This table shows which processes
should be running.

TABLE 12–3   Identity Synchronization for Windows Processes

Java Process Class Name	Component	When Present
com.sun.directory.wps.watchdog.server.WatchDog	System Watchdog	Always
com.sun.directory.wps.centrallogger.CentralLoggerManager	Central Logger	Only where Core is installed
com.sun.directory.wps.manager.SystemManager	System Manager	Only where Core is installed
com.sun.directory.wps.controller.AgentHarness	Connector	One for each connector installed

If the expected number of processes are not running, then issue the following commands to
restart all Identity Synchronization for Windows processes.

```
# /etc/init.d/isw stop
# /etc/init.d/isw start
```

If the WatchDog process is running, but the expected number of java.exe processes are not
running, then see the "Examining WatchList.properties" section to verify that all components
were installed properly.

Like other system components, the Sun Java System Directory Server Plug-in sends log records
over the bus that are managed by the central logger for end-user viewing. However, the Plug-in
also logs some messages that may not show up over the bus (for instance when the
subcomponent cannot contact the connector). In this case the log messages only show up in the
Plug-in's logs directory on the file system, which should look something like the following:

serverroot/isw/logs/SUBC

Because the Plug-in runs with the Directory Server process, there could potentially be a
problem for the Plug-in's ability to write into its logs directory. This happens if the directory
server runs as a different user than the owner of the logs directory. In this case, it may be
necessary to give the Plug-in permission explicitly by changing the directories permission or
owner using native operating system commands.

On Windows

Using the Service control panel, check that the "Sun Java System Identity Synchronization for Windows service is started. If it is not started, then Identity Synchronization for Windows is not running on that machine, and should be started. If the service is started, then verify using the Task Manager that pswwatchdog.exe (the Watchdog process) is running and that the expected number of java.exe processes are running:

- One for the Message Queue broker only if the Core is installed
- One for the System Manager only if the Core is installed
- One for the Central Logger only if the Core is installed
- One for each Connector installed on that machine

Note – There might be other active java processes, such as the Directory Server Console. If pswwatchdog.exe is not running, then restart the "Sun Java System Identity Synchronization for Windows" service. If it is running but the expected number of java.exe processes are not running, then see to verify that all components were installed properly.

Examining WatchList.properties

On each machine where a Identity Synchronization for Windows component is installed, the isw-*machine_name*/resources/WatchList.properties file enumerates the components that should run on that machine. The process.name[n] properties name the components that should be running.

On machines where Core is installed, WatchList.properties will include entries for the Central Logger and System Manager:

```
process.name[1]=Central Logger
process.name[2]=System Manager
```

On machines where connectors are installed, WatchList.properties will include a separate entry for each connector. The process.name property is the connector ID:

```
process.name[3]=CNN100
process.name[4]=CNN101
```

If there is a mismatch between the entries in WatchList.properties and the actively running processes, then restart the Identity Synchronization for Windows daemon or service.

If there are fewer than expected entries in WatchList.properties (e.g. only one connector entry even though two were installed), then examine the installation logs for possible installation failures.

- **On Solaris systems**: Installation logs are written to `/opt/SUNWisw`
- **On Linux Systems**: Installation logs are written to `/var/opt/sun/isw/logs`
- **On Windows systems**: Installation logs are written to the `%TEMP%` directory, which is a subdirectory of the `Local Settings` folder located under

  ```
  C:\Documents and Settings\Administrator
  ```

Note – On some Windows systems (such as Windows 2000 Advanced Server), the `Local Settings` folder is a hidden folder.

▼ To View Hidden Folders and the `Temp` subdirectory

1   **Open your Windows Explorer and select Tools → Folder Options from the menu bar.**

2   **When the Folder Options dialog box is displayed, select the View tab and enable the Show Hidden Files option.**

Troubleshooting Subcomponents

▼ To Troubleshoot Subcomponents in Your Deployment

1   **Have all subcomponents been installed?**

Subcomponent installation must be done after the connector is installed:

- For Active Directory Connectors, no subcomponents are installed.
- For Sun Java System Directory Server Connectors, the Directory Server Plug-in must be installed on the Sun Java System Directory Server being synchronized.
- For Windows NT Connectors, the Windows Change Detector and Password Filter subcomponents must be installed on the primary domain controller for each Windows NT domain being synchronized. These two subcomponents are installed together after the Windows NT Connector has been installed.

> **Note –** For the Windows NT SAM Change Detector subcomponent to be effective, you must turn on the NT audit log. Select Start → Programs → Administrative Tools → User Manager, and then select Policies → Audit Policies. Select Audit These Events and then both the Success and Failure boxes for User and Group Management.
>
> Select Event Log Settings in the Event Viewer → Event Log Wrapping, and then select Overwrite Events as Needed.

2   Have the subcomponent post-installation steps been followed?

After the Directory Server Plug-in has been installed at the Sun Java System Directory Server, the server must be restarted. After the NT Change Detector and Password Filter have been installed on the primary domain controller, the server must be restarted.

3   Are the subcomponents running?

Is the Directory Server where the Plug-in was installed running? Is the Primary Domain Controller where the Change Detector and Password Filter were installed running?

4   Have the subcomponents established a network connection to the connector?

On the machine where the connector is running, verify that the connector is listening for the subcomponent's connection by running `netstat -n -a`. The following examples show the results of this command for three different scenarios. (The connector was configured to listen on port 9999.)

a.   The connector is listening for incoming connections, and the subcomponent has successfully connected, which is the expected result:

```
# netstat -n -a | grep 9999
*.9999
*.*    0   0 65536    0 LISTEN
12.13.1.2.44397 12.13.1.2.9999
73620 0 73620    0 ESTABLISHED
12.13.1.2.9999
12.13.1.2.44397 73620 0 73620
0 ESTABLISHED
```

b.   The connector is listening for incoming connections, but the subcomponent has not connected:

```
# netstat -n -a | grep 9999
*.9999
*.*    0   0 65536    0 LISTEN
```

After verifying that the subcomponent is running, examine the subcomponent's local logs for potential problems.

c. **The connector is not listening for incoming connections:**

```
# netstat -n -a | grep 9999
#
```

Verify that the correct port number was specified. Verify that the connector is running and is in the READY state. Examine the connector's local logs for potential problems.

Troubleshooting Message Queue

Verify that the Sun Java System Message Queue broker is running. Issuing a telnet command to the machine and port where the Message Queue broker is running will return a list of the active Message Queue services:

```
# telnet localhost 7676
Trying 127.0.0.1...
Connected to localhost.
Escape character is \q^]\q.
101 psw-broker 3.0.1
cluster tcp CLUSTER 32914
admin tcp ADMIN 32912
portmapper tcp PORTMAPPER 7676
ssljms tls NORMAL 32913
jms tcp NORMAL 32911
Connection closed by foreign host.
```

If the "ssljms tcp NORMAL" service is not listed in the output, then examine the Message Queue logs for potential problems.

- If the Core was installed on Solaris, then the Message Queue broker's log is:
  /var/imq/instances/psw-broker/log/log.txt

- If the Core was installed on Linux, then the broker's log is:
  /var/opt/sun/mq/instances/isw-broker/log/log.txt

- If the Core was installed on Windows, then the broker's log is:
  installation_root**isw-***machine_name***imq****var****instances****isw-broker****log****log.txt**

  If the telnet command fails, then either the broker is not running or the wrong port was specified. Check the port number in the broker's log. The broker's port is specified in the following line

```
[13/Mar/2003:18:17:09 CST] [B1004]:
'Starting the portmapper service using tcp
[ 7676, 50 ]
with min threads 1 and max threads of 1'
```

If the broker is not running, then it can be started on Solaris and on Linux by running /etc/init.d/imq start and on Windows by starting the iMQ Broker Windows service.

If you are installing Message Queue on Solaris 8, and you will be running `mquinstall` to install all of the packages, be sure to set `IMQ_JAVAHOME` *before* running `mqinstall` to ensure the software picks the correct version of Java.

If you have not yet installed Core, you do not have to set `IMQ_JAVAHOME` because the Identity Synchronization for Windows installation program tells the Message Queue broker which JVM to use.

Troubleshooting Broker Configuration Directory Communication

The Message Queue broker authenticates clients against the Directory Server that stores the Identity Synchronization for Windows configuration. If the broker cannot connect to this Directory Server, no clients will be able to connect to the Message Queue, and the broker log will mention some `javax.naming` exception, such as "`javax.naming.CommunicationException`" or "`javax.naming.NameNotFoundException`".

If a `javax.naming` exception occurs, do the following:

- Verify that all `imq.user_repository.ldap` properties in `/var/imq/instances/isw-broker/props/config.properties` have the correct values. If any of these is incorrect, stop the Message Queue broker, correct and save the file, and restart the broker. The directory server host name must be resolvable from the broker's machine.

- Verify that the `imq.user_repository.ldap.password` property in `/etc/imq/passfile` is correct.

- In some cases, the broker cannot search for entries if the root suffix contains spaces.

Troubleshooting Broker Memory Settings

During normal operation, the Message Queue broker consumes a modest amount of memory. However, during `idsync resync` operations, the broker's memory requirements increase. If the broker reaches its memory limit, undelivered messages will accumulate, the `idsync resync` operation will slow down dramatically (or stop completely), and Identity Synchronization for Windows might be unresponsive afterwards.

When the broker enters a low-memory state, the following messages will appear in its log:

```
[03/Nov/2003:14:07:51 CST] [B1089]:
In low memory condition,
Broker is attempting to f
ree up resources [03/Nov/2003:14:07:51 CST] [B1088]:
Entering Memory State [B0024]:
```

```
RED  from previous state [B0023]:
ORANGE - current memory is 1829876K,
90% of total memory
```

To avoid this situation:

- Increase the broker's memory limit to 1 or 2 GB, as explained in *Sun Java System Directory Server Enterprise Edition 6.0 Release Notes*.

- During an `idsync resync` operation, keep the log level set to INFO. Changing the log level to FINE or higher increases the load at the broker as more log messages are sent to the central logger.

- Run `idsync resync` for a single Synchronization User List at a time.

▼ To Recover from Out of Memory Problems by the Broker

1. **Verify that the broker has a backlog of undelivered messages by examining its persistent message store in the appropriate directory.**

   - **On Solaris**: /var/imq/instances/psw-broker/filestore/message/

   - **On Linux**: /var/opt/sun/mq/instances/isw-broker/filestore/message/

   - **On Windows**: *installation_root*\isw-*machine_name*\imq\var\instances\isw-broker\filestore\message\

2. **Each file in this directory contains a single undelivered message. If there are more than 10000 files in this directory, then the broker has a backlog of messages. Otherwise, there is another problem with the broker.**

3. **The message backlog is probably only log files related to an** `idsync resync` **operation and you can safely remove them.**

4. **Stop the Message Queue broker as described in**

5. **Remove all files in the persistent message store. The easiest way to remove these files is by recursively removing the** `message/` **directory and then re-creating it.**

6. **Restart the Message Queue broker.**
   Follow the instructions in this section to ensure the broker does not run out of memory again.

Troubleshooting SSL Problems

When diagnosing problems with SSL, also see Chapter 4 and Chapter 10. This section contains:

- "SSL Between Core Components" on page 277
- "SSL between Connectors and Directory Server or Active Directory" on page 277
- "SSL between the Directory Server Plug-in and Active Directory" on page 281

SSL Between Core Components

The Identity Synchronization for Windows installation program cannot verify that the SSL port provided during Core installation is correct. If you type the SSL port incorrectly during Core installation, then the Core components will not be able to communicate properly. You may not notice a problem until you try to save your configuration for the first time. The Console will display the following warning:

```
The configuration was successfully saved,
however, the System Manager could not be
notified of the new configuration.
```

The system manager logs will have the following entry:

```
[10/Nov/2003:10:24:35.137 -0600] WARNING 14
example  "Failed to connect
to the configuration directory because "Unable to connect: (-5981)
Connection refused by peer."
Will retry shortly."
```

In this situation, uninstall the Core and install it again with the correct SSL port number.

SSL between Connectors and Directory Server or Active Directory

If a connector is unable to connect over SSL to the Directory Server or Active Directory, then this message will appear in the central error log:

```
[06/Oct/2006:14:02:48.911 -0600]
WARNING 14  CNN100 host1
"failed to open connection
to ldaps://host2.example.com:636."
```

Open the Console and check the Specifying Advanced Security Options panel (see "Creating an Active Directory Source" on page 156).

Untrusted Certificates

More information will be available in the central audit log. For example, if the LDAP server's SSL certificate is not trusted this message will be logged

```
[06/Oct/2006:14:02:48.951 -0600] INFO
14  CNN100 host1  "failed to open connection to
ldaps://host2.example.com:636, error(91):
Cannot connect to the LDAP server,
reason: SSL_ForceHandshake failed:
(-8179) Peer's Certificate issuer
is not recognized."
```

In most situations, the CA certificate has not been added to the connector's certificate database. This can be confirmed by running the `certutil` program that ships with the Directory Server.

Note – Certificate management utilities such as certutil are provided in the SUNWtlsu package, which is not bundled with the Directory Server. (You can download this package from Sun Microsystems at no cost.)

After downloading this package, you will find certutil in:

```
/usr/sfw/bin/certutil
```

In this example, the certificate database contains no certificates:

```
# /usr/sunone/servers/shared/bin/certutil
 -L -d /usr/sunone/servers/
 isw-host1/etc/CNN100
Certificate Name            Trust Attributes
p    Valid peer
P    Trusted peer (implies p)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

In the following example, the certificate database contains only the Active Directory CA certificate:

```
# /usr/sunone/servers/shared/bin/certutil -L -d
/usr/sunone/servers/ isw-host1/etc/CNN100
Certificate Name                            Trust Attributes
example.com CA                               C,c,
p    Valid peer
P    Trusted peer (implies p)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

As shown here, the trust flags of the CA certificate must be " C, ,". If the certificate exists and the trust flags are set properly, but the connector still cannot connect, then first verify that the connector was restarted after adding the certificate, and then use the ldapsearch command that ships with the Sun Java System Directory to help diagnose the problem. If ldapsearch does not accept the certificate, then neither will the connector. For example, ldapsearch can reject certificates if they are not trusted

```
# /usr/sunone/servers/shared/bin/ldapsearch
```

```
-Z -P /usr/sunone/ servers/isw-host1/etc/CNN100
-h host2 -b "" -s base "(objectclass=*)
"ldap_search: Can't contact LDAP server
SSL error -8179
Peer's Certificate issuer is not recognized.)
```

The -P option directs ldapsearch to use connector CNN100's certificate database for SSL certificate validation. After the correct certificate is added to the connector's certificate database, verify that ldapsearch accepts the certificate, and then restart the connector.

Mismatched Hostnames

When Identity Synchronization for Windows tries to establish SSL connections (with the trust all certificates setting disabled), the Identity Synchronization for Windows' Connectors verify that the server's hostname matches the hostname in the certificate that is presented by the server during the SSL negotiation phase. If the hostnames do not match, the connector will refuse to establish the connection.

The directory source hostname in the Identity Synchronization for Windows configuration must always match the hostname embedded in the certificate used by that directory source.

You can use ldapsearch to verify that the hostnames match, as follows:

```
/var/mps/serverroot/shared/bin/ldapsearch.exe
-Z -P /var/opt/SUNWisw/etc/CNN100 -3
-h host2.example.com -p 636
-s base -b "" "(objectclass=*)"
```

If there is a mismatch between the hostname in the command line (host2.example.com) and the hostname embedded in the certificate, then the following error message is displayed:

```
ldap_search: Can't contact LDAP server
SSL error -12276
(Unable to communicate securely with peer: requested
do main name does not match
the server's certificate.)
```

If the hostnames match, the ldapsearch command is successful and displays the contents of the root DSE.

Expired Certificates

If the server's certificate has expired, this message will be logged

```
[06/Oct/2006:14:06:47.130 -0600]
INFO    20  CNN100 host1
"failed to open connection to ldaps://host2.example.com:636,
```

```
error(91): Cannot connect to the LDAP server,
reason: SSL_ForceHandshake failed:
(-8181) Peer's Certificate has expired."
```

In this case, the server must be issued a new certificate.

SSL between the Directory Server Plug-in and Active Directory

By default, Directory Server does not communicate with Active Directory over SSL when performing on-demand password synchronization. If the default is overridden to protect this communication with SSL, then the Active Directory CA certificate must be added to the Directory Server certificate database of each master replica as described in Chapter 3 If this certificate is not added, users will fail to bind to Directory Server with the error "DSA is unwilling to perform.", and the Plug-in's log (for example, isw-*hostname* /logs/SUBC100/pluginwps_log_0.txt) will report:

```
[06/Nov/2006:15:56:16.310 -0600]
INFO    td=0x0376DD74 logCode=81
ADRepository.cpp:310
"unable to open connection to Active Directory server
at ldaps://host2.example.com:636, reason: "
```

In this situation, you must add the Active Directory CA certificate to Directory Server's certificate database and restart Directory Server.

Troubleshooting Controller Problems

Some counters will not be reset when you restore an Active Directory domain controller from back-up files.

To ensure all counters will be reset appropriately, resynchronize all users after restoring the an Active Directory domain controller.

Identity Synchronization for Windows
Appendixes

A

◆ ◆ ◆   **APPENDIX A**

Using the Identity Synchronization for Windows Command Line Utilities

Identity Synchronization for Windows enables you to perform a variety of tasks from the command line. This appendix explains how to execute the Identity Synchronization for Windows command line utilities to perform different tasks. The information is organized into the following sections:

Common Features

The Identity Synchronization for Windows command line utilities share the following features:

Common Arguments to the Idsync Subcommands

This section describes the arguments (options) that are common to most of the command line utilities. The information is organized into the following tables:

- Common Arguments to the Idsync Subcommands: Describes the following arguments, which are common to all of the idsync subcommands (*except prepds*) and migration tools.

  -D *bind-DN* -w *bind-password* | - [-h *Configuration Directory-hostname*]
  [-p *Configuration Directory-port-no*] [-s *rootsuffix*] [-Z] [-P *cert-db-path*]
  [-m *secmod-db-path*]

> **Note –** Brackets [ ] indicate optional arguments.
>
> The Identity Synchronization for Windows installation program automatically writes default values to the -h, -p, -D, and -s arguments based on the information you provide during installation. However, you can specify a different value on the command line to override a defaulted value.
>
> To support multibyte characters, Identity Synchronization for Windows base64-encodes the default values for -s *rootsuffix* and -D *bind-DN* in the command line interface (CLI) environment file. The rootsuffix default should not be changed. The bind DN default can be overridden on the command line or updated with the appropriate base64-encoded value in the CLI environment file.

- Common Arguments for Accessing the Configuration Directory Server using SSL: Describes optional arguments that provide information about securely accessing the Configuration Directory Server using Secure Socket Layer (SSL). These arguments are also common to all of the idsync subcommands and the migration tools.

- Common Arguments Related to Configuration Directory: Describes arguments related to the configuration directory. These arguments are common to two or more idsync subcommands and migration tools.

TABLE A–1 Arguments Common to All Subcommands

Argument	Description
-h *Configuration Directory-hostname*	Specifies the configuration directory hostname. This argument defaults to the values specified during Core installation.
-p *Configuration Directory-port*	Specifies the configuration directory LDAP port number.
-D *bind-DN*	Specifies the configuration directory bind distinguished name (DN). This argument defaults to the values specified during Core installation.
-w *bind-password* | -	Specifies the configuration directory bind password. The - value reads the password from standard input (STDIN).
-s *rootsuffix*	Specifies the configuration directory rootsuffix. Where rootsuffix is a distinguished name such as dc=example,dc=com.This argument defaults to the values specified during Core installation.
-q *configuration_password* | -	Specifies the configuration password. The - value means the password will be read from standard input (STDIN). This argument is *mandatory* for all subcommands except prepds.

TABLE A–2 SSL-Related Arguments Common to All Subcommands

Argument	Description
-Z	Specifies that SSL be used to provide secure communication. Provides certificate-based client authentication when connecting to the configuration directory accessing the command line interface or the preferred/secondary Directory Servers.
-P *cert-db-path*	Specifies the path and file name of the client's certificate database.
	This certificate database must contain the CA certificate used to sign the Directory Server's certificate database.
	If you specify -Z but do not use -P, the *cert-db-path* defaults to *current-working-directory* /cert8.db.
	Note: If Identity Synchronization for Windows does not find the certificate database file in the specified directory, the program creates an *empty* database in that directory, which consists of three files: cert8.db, key3.db, and secmod.db.
-m *secmod-db-path*	Specifies the path to the security module database. For example:
	/var/Sun/MPS/slapd-*serverID* /secmod.db
	Specify this argument only if the security module database is in a different directory than the certificate database itself.

TABLE A–3 Configuration Directory Arguments

Argument	Description
-a *ldap_filter* Use with forcepwchg and resync subcommands	Specifies the LDAP filter to use when retrieving users from the source SULs, and allows the operation to retrieve a focused subset of users from the directory source, prior to determining whether the users fall within the specified SULs.
-f *filename* Use with export10cnf, importcnf, and resync subcommands	Specifies the name of a Configuration XML Document file.
-n Use with forcepwchg, importcnf, and resetconn subcommands	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Entering Passwords

Wherever a password argument is required (such as -w *bind-password* or -q *configuration_password*), you can use the " **-**" argument to tell the password program to read the password from STDIN.

If you use the "**-**" value for multiple password options, idsync will prompt you for passwords based on the arguments' order.

In this case, the program would expect the *bind-password* first, and then for the *configuration-password*.

Getting Help

You can use one of the following commands to display usage information about idsync or any of its subcommands in the command Console:

- **-help**
- **--help**
- **-?**

For usage information

- About idsync (including a list of valid subcommands), type one of the preceding help options at a command prompt and click Return.
- About a subcommand, type the subcommand followed by a help option at a command prompt and click Return.

Using the idsync command

You use the idsync command and subcommands to execute the Identity Synchronization for Windows command line utility.

Note – The idsync command converts all DN-valued arguments (such as bind DN or suffix name) from the character set specified for that window to UTF-8 before sending the arguments to Directory Server.

Do not use backslashes as escape characters in suffix names.

To specify UTF-8 characters on Solaris and on Linux, your terminal window must have a locale based on UTF-8. Make sure that the environmental variable's LC_CTYPE and LANG.are set correctly.

Unless specifically noted otherwise, you can run the idsync command with subcommands using either of the following methods:

- **From Solaris:**
    1. Open a terminal window and **cd** to the /opt/SUNWisw/bin directory.
    2. Type the idsync command with one subcommand, as follows

idsync *subcommand*

- **From Linux:**
  1. Open a terminal window and **cd** to the /opt/sun/isw/bin directory.
  2. Type the idsync command with one subcommand, as follows

     idsync *subcommand*

- **From Windows:**
  1. Open a Command Window and **cd** to the *install_path*\isw-*hostname*\bin directory.
  2. Type the idsync command with one subcommand, as follows

     idsync *subcommand*

     "Using the idsync command" on page 288 lists all of the idsync utility subcommands
     and their purpose:

TABLE A–4   Quick Reference to idsync Subcommands

Subcommand	Purpose
certinfo	Displays certificate information based on your configuration and SSL settings (see "Using certinfo" on page 290)
changepw	Changes the Identity Synchronization for Windows configuration password (see "Using changepw" on page 290)
importcnf	Imports an exported Identity Synchronization for Windows version 1.0 configuration XML document (see "Using importcnf" on page 292)
prepds	Prepares a Sun Java System Directory Server source for use by Identity Synchronization for Windows (see "Using prepds" on page 292)
printstat	Displays a list of steps you must perform to complete the installation/configuration process. Also provides the status of installed connectors, the system manager, and the Message Queue (see "Using printstat" on page 296)
resetconn	Resets connector states in the configuration directory to *uninstalled* (see "Using resetconn" on page 297)
resync	Links and resynchronizes existing users or groups and pre-populates directories as part of the installation process (see "Using resync" on page 297)
groupsync	Synchronizes group information between users and groups from one directory source to another (see "Using groupsync" on page 300)
accountlockout	Synchronizes account lockout and unlockout between Directory Server and Active Directory directory sources (see "Using accountlockout" on page 300)
dspluginconfig	Configures and unconfigures Directory Server plugin on a specified host (see "Using dspluginconfig" on page 301)

TABLE A–4	Quick Reference to idsync Subcommands	*(Continued)*
Subcommand	**Purpose**	
startsync	Starts synchronization (see "Using startsync" on page 301)	
stopsync	Stops synchronization (see "Using stopsync" on page 302)	

Using certinfo

You can use the certinfo subcommand to display certificate information based on your configuration and SSL settings. This information can help you determine which certificates must be added for each connector and/or Directory Server Plug-in certificate database.

To display certificate information, open a terminal window (or Command Window) and type the **idsync certinfo** command as follows:

```
idsync certinfo [bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z]
[-P cert-db-path] [-m secmod-db-path]
```

Note – Because the certinfo subcommand does not have access to the connectors' and Directory Server's certificate databases, some of the required steps it lists might have already been performed.

For example:

```
idsync certinfo -w admin-password -q configuration-password
```

Note – For detailed information about the certinfo arguments, review "Common Arguments to the Idsync Subcommands" on page 285.

Using changepw

You can use the changepw subcommand to change the Identity Synchronization for Windows configuration password.

▼ To Change the Configuration Password for Identity Synchronization for Windows:

1 **Stop all Identity Synchronization for Windows processes (for example, System Manager, Central Logger, Connectors, Console, Installers/Uninstallers).**

2   **After stopping all the processes, back up the ou=Services tree by exporting the configuration directory to** ldif.

3   **Type the** idsync changepw **command as follows:**

```
idsync changepw [-D bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password
[-Z] [-P cert-db-path] [-m secmod-db-path]
-b new password | - [-y]
For example:

idsync changepw -w admin password -q old config password -b -q new config password
```

The following arguments are unique to changepw:

Argument	Description
-b *password*	Specifies a new configuration password. The - value reads the password from standard input (STDIN).
[-y]	Does not prompt for command confirmation.

4   **Respond to the messages that display in the terminal window. For example,**

```
Are you sure that want to change the configuration password (y/n)? yes
Before restarting the system -
you must edit the $PSWHOME/resources/SystemManagerBootParams.cfg file
and change the 'deploymentPassword' to the new value.

SUCCESS
```

5   **You must modify the** SystemManagerBootParams.cfg **file before restarting the system.**

The SystemManagerBootParams.cfg file in $PSWHOME\resources (where $PSWHOME is the *isw-installation directory*) contains the configuration password the system manager uses to connect to the configuration directory.

For example, you would change the password value as follows:

From: *Parameter name="manager.configReg.deploymentPassword" value="* **oldpassword**"/

To: *Parameter name="manager.configReg.deploymentPassword" value=* "**newpassword** "/

6   **If the program reports any errors, restore the configuration directory using the** ldif **from <u>"Using changepw" on page 290</u> and then try again. The most likely reason for an error is that the Directory Server hosting the configuration directory became unavailable during the password change.**

Using importcnf

After installing Core (Chapter 5), use the idsync importcnf subcommand to import your exported Identity Synchronization for Windows version 1.0 or 1.1 (SP1) configuration XML file, which contains Core configuration information.

To import your version 1.0 configuration XML file, open a terminal window (or Command Window) and type the **idsync importcnf** command as follows:

```
idsync importcnf [-D bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z] [-P cert-db-path]
[-m secmod-db-path] -f filename [-n]
```

For example:

```
idsync importcnf -w admin_password -q configuration_password -f "MyConfig.cfg"
```

The following arguments are unique to importcnf:

TABLE A–5 idsync importcnf Arguments

Argument	Description
-f filename	Specifies the name of your configuration XML document.
-n	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Note – For detailed information about other importcnf arguments, review "Common Arguments to the Idsync Subcommands" on page 285.

After importing the version 1.0 configuration XML file, you must run prepds on all Directory Server sources configured for synchronization, (see "Using prepds" on page 292 connectors and subcomponents.

Using prepds

You use the console or prepds subcommand to prepare a Sun Java System Directory Server source for use by Identity Synchronization for Windows. You must run prepds before installing the Directory Server Connector.

Running the idsync prepds subcommand applies the appropriate ACI to the cn=changelog entry, which is the root node of the Retro-Changelog database.

If you are preparing a *preferred master* Directory Server for use by Identity Synchronization for Windows, you must provide *Directory Manager* credentials.

The Directory Manager user is a special user on Directory Server who has full rights anywhere inside the Directory Server instance. (ACI does not apply to Directory Manager users.)

For example, only the Directory Manager can set the access control for the Retro-Changelog database, which is one of the reasons why Identity Synchronization for Windows requires Directory Manager credentials for the preferred master server.

Note – If you re-create the Retro-Changelog database for the preferred Sun directory source for any reason, the default access control settings will not allow the Directory Server Connector to read the database contents.

To restore the access control settings for the Retro-Changelog database, run idsync prepds or click the Prepare Directory Server button after selecting the appropriate Sun directory source in the Console.

You can configure your system to automatically remove (or *trim*) Change-log entries after a specified period of time. From the command line, modify the nsslapd-changelogmaxage configuration attribute in cn=Retro Changelog Plug-in, cn=plugins, cn=config:

nsslapd-changelogmaxage: *IntegerTimeunit*

Where:

- **Integer** is a number.
- **Timeunit** is s for seconds, m for minutes, h for hours, d for days, or w for weeks. (There should be no space between the Integer and Timeunit variables.)

    For example, nsslapd-changelogmaxage: 2d

    For more information, see the "Managing Replication" chapter in the Sun Java System Directory Server 5 2004Q2 Administration Guide.
- You can use *Administrative* credentials to prepare a *secondary* server.

Be sure to plan your Identity Synchronization for Windows configuration *before* running idsync prepds because you must know which hosts and suffixes you will be using.

Running idsync prepds on a Directory Server suffix where the Directory Server Connector and Plug-in are already installed, configured, and synchronizing will result in a message asking you to install the Directory Server Connector. Disregard this message.

To prepare a Sun Java System Directory Server source, open a terminal window (or a Command Window) and type the **idsync prepds** command as follows:

For single host:

idsync prepds [-h *<hostname>*] [-p *<port>*] [-D *<Directory Manager DN>*] -w *<password>*
-s *<database suffix>* [-x] [-Z] [-P *<cert db path>*] [-m *<secmod db path>*]

For multiple hosts:

idsync prepds -F *<filename of Host info>* -s *<root suffix>* [-x] [-Z]
[-P *<cert db path>*][-m *<secmod db path>*] [-3]

For example:

idsync prepds -D "cn=Directory Manager" -w *preferred master password*
-h *preferred-host* -p 389 -s dc=example,dc=com -j "secondary host" -r 389
-E "cn=Administrator" -u *secondary master password* -s dc=example,dc=com

---

Note – The -h, -p, -D, -w, and -s arguments are redefined (as described in the following table) for the prepds subcommand only. In addition, the -q argument does not apply.

"Using prepds" on page 292 describes the arguments that are unique to idsync prepds.

TABLE A–6   prepds Arguments

Argument	Description
-h *name*	Specifies the DNS name of the Directory Server instance serving as the preferred host.
-p *port*	Specifies port number for Directory Server instance serving as preferred host. (*Default is 389.*)
-j *name* (*optional*)	Specifies the DNS name of the Directory Server instance serving as the secondary host (applicable in a Sun Java System Directory Server 5 2004Q2 multimaster replicated (MMR) environment).
-r *port* (*optional*)	Specifies a port for the Directory Server serving as the secondary host (applicable in a Sun Java System Directory Server 5 2004Q2 multimaster replicated (MMR) environment). (*Default is 389.*)
-D *dn*	Specifies the distinguished name of the Directory Manager user for the preferred host.
-w *password*	Specifies a password for the Directory Manager user for the preferred host. The - value reads the password from standard input (STDIN).
-E *admin-DN*	Specifies the distinguished name of the Directory Manager user for the secondary host.
-u *password*	Specifies a password for the Directory Manager user for the secondary host. The - value reads the password from standard input (STDIN).

TABLE A–6 prepds Arguments *(Continued)*

Argument	Description
-s *rootsuffix*	Specifies the root suffix to use for adding an index (root suffix where you will be synchronizing users). **Note**: The database name of the Preferred and Secondary hosts may vary, but the suffix will not. Consequently, the program can find the database name of each host and use it to add the indexes.
-x	Does not add equality and presence indexes for dspswuserlink attribute to the database.
-F *filename of Host info*	Specifies the filename containing the host information in case of multiple hosts environment.

If you are running `idsync prepds` in a replicated environment, (for example, where you have a preferred master, a secondary master, and two consumers), you only need to run `idsync prepds` once for the preferred and secondary masters.

▼ **To run** `idsync prepds`

1 **Ensure that Directory Server replication is up and running (if applicable.)**

2 **Run** `idsync prepds` **from the Console or from the command line, for example:**

    idsync prepds -h M1.example.com -p 389 -j M2.example.com -r 389.

Running the `idsync prepds` command on M1 accomplishes the following:

- Enables and extends the RCL to capture more attributes (dspswuserlink and so forth)

  RCL is required on M1 only.

- Extends schema.

- Adds uid=pswconnector, *suffix* user with ACIs.

- Adds indexes to the dspswuserlink attribute, which puts Directory Server in read-only mode temporarily until the indexing is done.

  You can add indexes later to avoid downtime, but you must add indexes *before* installing the Directory Server Connector.

Adds indexes on M2.

Note –

- Replication ensures that Identity Synchronization for Windows copies schema information and the `uid=pswconnector` from the preferred master to the secondary master and both consumers.
- You must install the Directory Server Connector once. You must install the Directory Server Plug-in in *all* directories.
- Indexing is required on the preferred and the secondary masters only. (Replication does not push the indexing configuration from the preferred master to the secondary master.)

Using printstat

You can use the `printstat` subcommand to:

- Display a list of the remaining steps you have to perform to complete the installation and configuration process.
- Print the status of installed connectors, the system manager, and the Message Queue.

  Possible status settings include:

  - **Uninstalled.** The connector is not installed.
  - **Installed.** The connector is installed, but not ready for synchronization because it has not received its runtime configuration yet.
  - **Ready.** The connector is ready for synchronization, but is not synchronizing any objects yet.
  - **Syncing.** The connector is synchronizing objects.

    To print the status of installed Connectors, the System Manager, and the Message Queue open a terminal window (or a Command Window) and enter the **idsync printstat** command as follows:

```
idsync printstat [-D bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z]
[-P cert-db-path] [-m secmod-db-path]
```

   For example:

```
idsync printstat -w admin password -q configuration password
```

Using resetconn

You can use the `resetconn` subcommand to reset connector states in the configuration directory to *uninstalled*. For example, if a hardware failure prevents you from uninstalling a connector, use `resetconn` to change the connector's status to uninstalled so you can reinstall that connector.

Note – The `resetconn` subcommand is intended to be used only in the event of hardware or uninstaller failures.

To reset the state of connectors from the command line, open a terminal window (or a Command Window) and type the **idsync resetconn** command as follows:

```
idsync resetconn [-D bind-DN] -w bind-password\> | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z] [-P cert-db-path]
[-m secmod-db-path] -e directory-source-name [-n]
```

For example:

```
idsync resetconn -w admin password -q configuration_password -e "dc=example,dc=com"
```

"Using prepds" on page 292 describes the arguments that are unique to `resetconn`:

TABLE A–7 idsync resetconn Arguments

Argument	Description
`-e` *dir-source*	Specifies the name of the directory source to reset.
`-n`	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Note – `idsync printstat` can be used to find directory source names.

For detailed information about the other `resetconn` arguments, review "Common Arguments to the Idsync Subcommands" on page 285.

Using resync

You can use the `resync` subcommand to bootstrap deployments with existing users. This command uses administrator-specified matching rules to

- Link existing entries

- Populate an empty directory with the contents of a remote directory

- Bulk-synchronize attribute values between two existing user populations

- Bulk-synchronize existing groups and the users associated with the groups (when the group synchronization feature is enabled).

Note – For more detailed information about linking and synchronizing users, see Chapter 3.

To resynchronize existing users and to pre-populate directories, open a terminal window (or a Command Window) and type the **idsync resync** command as follows:

```
idsync resync [-D bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z] [-P cert-db-path]
[-m secmod-db-path] [-n] [-f xml filename for linking] [-k] [-a ldap-filter]
[-l sul-to-sync] [-o Sun | Windows] [-c] [-x]
[-u][-i ALL_USERS | NEW_USERS | NEW_LINKED_USERS]
```

For example:

```
idsync resync -w admin password -q configuration_password
```

"Using resync" on page 297 describes the arguments that are unique to resync:

TABLE A–8 idsync resync Usage

Argument	Meaning
-f *filename*	Creates links between unlinked user entries using one of the specified XML configuration files provided by Identity Synchronization for Windows (see Appendix B)
-k	Creates links between unlinked users only (does not create users or modify existing users)
-a *ldap-filter*	Specifies an LDAP filter to limit the entries to be synchronized. The filter will be applied to the source of the resynchronization operation. For example, if you specify idsync resync -o Sun -a "uid=*" all Directory Server users that have a uid attribute will be synchronized to Active Directory.
-l *sul-to-sync*	Specifies individual Synchronization User Lists (SULs) to resynchronize
	Note: You can specify multiple SUL IDs to resynchronize multiple SULs or, if you do not specify any SUL IDs, the program will resynchronize all of your SULs.

TABLE A–8 idsync resync Usage   *(Continued)*

Argument	Meaning
-o (Sun | Windows)	Specifies the source of the resynchronization operation ■ **Sun**: Sets attribute values for Windows entries to corresponding attribute values in Sun Java System Directory Server directory source entries. ■ **Windows**: Sets attribute values for Sun Java System Directory Server entries to corresponding attribute values in Windows directory source entries. (*Default is Windows.*)
-c	Creates a user entry automatically if the corresponding user is not found at destination ■ Randomly generates a password for users created in Active Directory or Windows NT ■ Automatically creates a special password value ((PSWSYNC) ***INVALID PASSWORD***) for users created in Directory Server (unless you specify the -i option)
-i (ALL_USERS | NEW_USERS | NEW_LINKED_USERS)	Resets passwords for user entries synchronized in the Sun directory sources, forcing password synchronization within the current domain for those users the next time the user password is required. ■ **ALL_USERS**: Forces on-demand password synchronization for all synchronized users ■ **NEW_USERS**: Forces on-demand password synchronization for newly created users only ■ **NEW_LINKED_USERS**: Forces on-demand password synchronization for all newly created and newly linked users
-u	Only updates the object cache. No entries are modified. This argument updates the local cache of user entries for a Windows directory source only, which prevents pre-existing Windows users from being created in Directory Server. If you use this argument, Windows user entries are not synchronized with Directory Server user entries. This argument is valid only when the resync source is Windows.
-x	Deletes all destination user entries that do not match a source entry.
-n	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Note –

- Run idsync resync with no arguments to view a usage statement.
- For detailed information about the resync arguments, review "Common Arguments to the Idsync Subcommands" on page 285.
- For more information about resynchronizing existing users, review Chapter 3.

  After running resync, check the resync.log file in the central audit log. If errors result, consult "Troubleshooting Checklist" on page 262

Using groupsync

You can use the groupsync subcommand to synchronize groups between Active Directory and Directory Server.

To enable or disable the Group Synchronization, type **idsync groupsync** command.

For example:

```
idsync groupsync -{e/d} -D <bind DN> -w <bind password> [-h <CD hostname>]
[-p <CD port no>] -s <rootsuffix> [-Z] -q <configuration password> -t <AD group type>
```

TABLE A–9 groupsync arguments

Argument	Meaning
-{e/d}	Select e for enabling , and d for disabling the group synchronization.
-t	Specifies the group type at Active Directory. For example, it can be selected as either of "distribution" or "security"

Using accountlockout

You can use the accountlockout subcommand to synchronize account lockout and unlockout between Active Directory and Directory Server.

To enable or disable the account lockout, type **idsync accountlockout** command.

For example:

```
idsync accountlockout -{e/d} -D <Directory Manager DN> -w <bind-password>
-h <Configuration Directory-hostname> -p <Configuration Directory-port-no>
-s <rootsuffix> [-Z] [-P <cert db path>] [-m <secmod db path>]
-q <configuration password> -t <max lockout attempts>
```

TABLE A–10 accountlockout arguments

Argument	Meaning
-{e/d}	Select e for enabling , and d for disabling the account lockout synchronization.
-t	Specifies the maximum number of lockout attempts that Active Directory Connector performs.

Using dspluginconfig

You can use the dspluginconfig subcommand to configure or unconfigure Directory Server plugin on a specified Directory Server data source.

To configure or unconfigure the Directory Server plugin, type **idsync dspluginconfig**command.

For example:

```
idsync dspluginconfig -{C/U} -D <bind DN> -w <bind password | ->
[-h <CD hostname>] [-p <CD port no>] [-s <configuration suffix>]
[-Z] [-P <cert db path>] [-m <secmod db path> ] [-d <ds plugin hostname>]
[-r <ds plugin port>] [-u <ds plugin user>] [-x <ds plugin user password>]
[-o <database suffix>]   [-q <configuration password | ->]
```

TABLE A–11 dspluginconfig arguments

Argument	Meaning
-{C/U}	Select C for configuring and U for unconfiguring the Directory Server plugin.
-d	Host name of the Directory Server data source where the plugin needs to be configured
-r	Port number of the Directory Server data source where the plugin needs to be configured
-u	Administrator of the Directory Server data source where the plugin needs to be configured
-x	Password of the administrator of the Directory Server data source where the plugin needs to be configured
-o	Data suffix of the Directory Server data source.

Using startsync

You can use the startsync subcommand to start synchronization from the command line.

To start synchronization, open a terminal window (or a Command Window) and type the **`idsync startsync`** command as follows:

```
idsync startsync [-D bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z]
[-P cert-db-path] [-m secmod-db-path]
```

For example:

```
idsync startsync -w admin password -q configuration_password
```

describes the arguments that are unique to `startsync`.

TABLE A–12   idsync startsync Arguments

Argument	Description
[-y]	Does not prompt for command confirmation.

Note – For detailed information about the other `startsync` arguments, review "Common Arguments to the Idsync Subcommands" on page 285.

Using stopsync

You can use the `stopsync` subcommand to stop synchronization from the command line.

To stop synchronization, open a terminal window (or a Command Window) and type the **`idsync stopsync`** command as follows:

```
idsync stopsync [-D bind-DN] -w bind-password | -
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]
[-s rootsuffix] -q configuration_password [-Z]
[-P cert-db-path] [-m secmod-db-path]
```

For example:

```
idsync stopsync -w admin password -q configuration_password
```

Note – For detailed information about the `stopsync` arguments, review "Common Arguments to the Idsync Subcommands" on page 285.

Using the forcepwchg Migration Utility

Users who change their passwords during migration will have different password in Windows NT and the Directory Server. You can use the forcepwchg utility to require a password change for users who changed their passwords during the Identity Synchronization for Windows version 1.0 to version 6.0 migration process.

Note – The forcepwchg utility ships with Windows packages only.

Before using forcepwchg you must verify the following:

- Be sure you do not configure the 7-bit check Plug-in in Directory Server to enforce 7-bit values for the userpassword attribute. Do this using the Directory Server console.
- Be sure that the client you are using for authentication translates the value from your locale to UTF-8 correctly. (For example, the -i option for the ldapsearch shipped with Directory Server).

▼ To Execute the forcepwchg Command line Utility

1 **Open a Command Prompt window and cd to the Windows** migration **directory on the host where you are performing the migration. (The Identity Synchronization for Windows 1.0 NT components such as connector, Change Detector DLL, and Password Filter DLL must be installed on the PDC host.)**

2 **From the** migration **directory, type**

    java -jar forcepwchg.jar [-n] [-a] [-t <
    *time_specification***\>]**

    For example,

    forcepwchg.jar -n -a forcepwchg.jar -t 33m

    describes the arguments that are unique to forcepwchg:

Option	Description
-n	Specifies *preview mode.* In preview mode, the utility prints out the names of all normal users except: ■ Built-in accounts (Administrator and Guest) if you specify the -a argument. ■ Users who changed passwords during the time specified using the -t argument. In preview mode, any user can execute forcepwchg. In non-preview mode, only the Administrator can execute forcepwchg.
-a	Requires all users (except Administrator and Guest) to change their passwords. You cannot use this argument if you are using the -t argument.
-t *time_specification*	Forces all users who changed passwords in the past *time specification* to change their passwords. Where *time specification* can have the following form: ■ *number*: Number of seconds (for example, -t 30) ■ *number* **m**: Number of minutes (for example, -t 25m) ■ *number* **h**: Number of hours (for example, -t 6h) For example, if you specify forcepwchg -t 6h, all users who changed passwords within the last six hours will be required to change their password again.
-?	Prints out usage information.

B

Identity Synchronization for Windows LinkUsers XML Document Sample

This appendix provides two sample XML configuration documents that you can use with the `idsync resync` subcommand to link existing users in your deployment.

Both of the following files are available in the `samples1` subdirectory where you installed Core:

- "Sample 1: linkusers-simple.cfg" on page 305 (an example of a common and simple configuration)

- "Sample 2: linkusers.cfg" on page 306 (a more-complex configuration example that shows the full power of specifying linking criteria)

You can modify the samples to suit your environment. Both files contain comments that explain how to modify the samples to link your users — including how to link users in multiple SULs.

Sample 1: linkusers-simple.cfg

```
<!--
    Copyright 2004 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms.
--\>
<!--
    This xml file is used to link Windows and
    Sun Directory Server users from the commandline.
    It is passed to the 'idsync resync'
    script as the -f option. This is a simple file
    that links users in the SUL1 synchronization user list
    that have the same login name, that is the Directory Server
    uid attribute matches the Active Directory
    samaccountname attribute. For more complex
    matching rules, see the linkusers.cfg sample.
--\>
```

```
<UserLinkingOperationList\>
 <UserLinkingOperation parent.attr="UserLinkingOperation"
            sulid="SUL1"\>
   <UserMatchingCriteria parent.attr="UserMatchingCriteria"\>
    <AttributeMap parent.attr="AttributeMap"\>
      <AttributeDescription parent.attr="SunAttribute"
              name="uid"/\>
      <AttributeDescription parent.attr="WindowsAttribute"
              name="samaccountname"/\>
      </AttributeMap\>
      </UserMatchingCriteria\>
   </UserLinkingOperation\>
</UserLinkingOperationList\>
```

Sample 2: linkusers.cfg

```
<?xml version ="1.0" encoding="UTF-8"?\>
<!--

    Copyright 2004 Sun Microsystems, Inc.
    All rights reserved
    Use is subject to license terms.
--\>
<!--
    This xml file is used to link Windows
    and Sun Directory Server users from
    the command line.  It is passed to the
    \qidsync resync\q script as the -f option.
--\>
<!--
    The following parameters allowLinkingOutOfScope:
    if true, then Windows users can be
    linked to Sun Directory Server users
    that are outside of the users\q Synchronization
    User List. Default is false.
--\>
<UserLinkingOperationList allowLinkingOutOfScope="false"\>


<!--
    UserLinkingOperation encapsulates the configuration
    of a single SUL to link. It includes the SUL ID
    and a list of attributes to match.
    A separate UserLinkingOperation must
    be specified for each SUL being linked.
--\>
<UserLinkingOperation parent.attr="UserLinkingOperation" sulid="SUL1"\>
```

```
<!--
    UserMatchingCriteria encapsulates a
    list of attributes that must match for a user to be linked. --\>
<!--
    For two users to match using this UserMatchingCriteria,
    they must have the same givenName and the same sn. --\>
<UserMatchingCriteria parent.attr="UserMatchingCriteria"\>
    <AttributeMap parent.attr="AttributeMap"\>
    <AttributeDescription parent.attr="SunAttribute" name="sn"/\>
    <AttributeDescription parent.attr="WindowsAttribute" name="sn"/\>
    </AttributeMap\>     <AttributeMap parent.attr="AttributeMap"\>
    <AttributeDescription parent.attr="SunAttribute" name="givenName"/\
    <AttributeDescription parent.attr="WindowsAttribute"
    name="givenName"/\>     </AttributeMap\></UserMatchingCriteria\>


<!--
    Multiple UserMatchingCriteria can be specified for a single SUL.
    They are treated as a logical OR.  In this example,
    the givenName\qs and sn\qs must match (see above)) OR
    (the employee(Number|ID) must match),
    for the user to be linked.  Notice that attribute
    that is specified, employeeNumber,
    is the name of the DS attribute. --\>
<!--
    This UserMatchingCriteria is commented out because
    employeeNumber is not an indexed attribute in DS.
    All attributes used in a UserMatchingCriteria
     should be indexed.
    <UserMatchingCriteria parent.attr="UserMatchingCriteria"\>
      <AttributeMap parent.attr="AttributeMap"\>
         <AttributeDescription parent.attr=
         "SunAttribute" name="employeeNumber"/\>
          <AttributeDescription parent.attr=
          "WindowsAttribute" name="employeeID"/\>
       </AttributeMap\>
    </UserMatchingCriteria\>
--\>
</UserLinkingOperation\>
<!--
    When multiple SULs are linked, a separate UserLinkingOperation
    is specified for each.
    As shown here, each UserLinkingOperation can use different
    UserMatchingCriteria: in this example, users in SUL2 are
    only linked if their sn and employeeNumber match.
    Note: this UserLinkingOperation is currently
    commented out because the example configuration
```

```
                      only has a single SUL.
            <UserLinkingOperation parent.attr="UserLinkingOperation" sulid="SUL2"\>
              <UserMatchingCriteria parent.attr="UserMatchingCriteria"\>
                <AttributeMap parent.attr="AttributeMap"\>
                    <AttributeDescription parent.attr="SunAttribute" name="sn"/\>
                    <AttributeDescription parent.attr="WindowsAttribute" name="sn"/\>
                 </AttributeMap\>
                    <AttributeMap parent.attr="AttributeMap"\>
                      <AttributeDescription parent.attr=
                      "SunAttribute" name="employeeNumber"/\>
                       <AttributeDescription parent.attr=
                       "WindowsAttribute" name="employeeID"/\>
                    </AttributeMap\>
              </UserMatchingCriteria\>
        </UserLinkingOperation\>
        --\>
        </UserLinkingOperationList\>
```

♦ ♦ ♦ **APPENDIX C**

Running Identity Synchronization for Windows Services as Non-Root on Solaris

You must have `root` privileges to install and to run Identity Synchronization for Windows services on Solaris and Red Hat systems.

However, after installing the product you can configure the software to run the program services as a non-`root` user.

Running Services as a Non-`root` User

Note – To run services as non-`root`, you must change the permissions for all directories under the Identity Synchronization for Windows instance directory. The *default* directory is `/var/opt/SUNWisw`.

▼ To Run services as a Non-`root` User

Although you must be `root` to install and to run Identity Synchronization for Windows services, you can configure the software to run the program services as a non-`root` user.

1   (Optional) Use the UNIX `useradd` command to create a user account for Identity Synchronization for Windows.

You also can use a `nobody` user to run services. The remaining examples in this procedure assume you created a user called `iswuser`.

2   To install a Sun Java System Directory Server Connector, you must choose a non-privileged port for the Connector during installation.

For example, ports larger than 1024 are acceptable. Port 1389 is recommended for LDAP when the server is running as a non-`root` user. Port 1636 is recommended for LDAP over SSL.

Note – You must execute all commands in the remaining steps as root.

---

3 After installing all components, execute the following command to stop Identity Synchronization for Windows:

```
/etc/init.d/isw stop
```

4 You must update the ownership of the instance directory. For example, if you installed the product in /var/opt/SUNWisw.

```
chown -R iswuser /var/opt/SUNWisw
```

```
chown -R iswuser /opt/SUNWisw
```

5 In a text editor, open the /etc/init.d/isw file and replace the following line:

```
"$EXEC_START_WATCHDOG" "$JAVA_PATH" "$INSTALL_DIR" "$CONFIG_DIR"
```

with the following:

```
su iswuser -c "$EXEC_START_WATCHDOG '$JAVA_PATH' '$INSTALL_DIR' '$CONFIG_DIR'"
```

6 Execute the following command to restart the service:

```
/etc/init.d/isw start
```

7 Execute the following command to verify that the components are running using the assigned user's userid:

```
ps -ef | grep iswuser
```

D

Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows

This appendix provides supplemental information about Synchronization User List (SUL) definitions and explains how to configure multiple domains. The information is organized as follows:

- "Understanding Synchronization User List Definitions" on page 311
- "Configuring Multiple Windows Domains" on page 313

Understanding Synchronization User List Definitions

Every Synchronization User List (SUL) contains two definitions — one to identify which Directory Server users to synchronize and the other to identify which Windows users to synchronize.

Each definition identifies which users in a directory to synchronize, which users to exclude from synchronization, and where to create new users.

Note – The objectclasses you select using the Identity Synchronization for Windows Console also determine which users will be synchronized. The program synchronizes only those users that have the selected objectclass, which includes any users that have a subclass of the selected objectclass.

For example, if you select the `organizationalPerson` objectclass, then Identity Synchronization for Windows will synchronize users with the `inetorgperson` objectclass because it is a subclass of the `organizationalPerson` objectclass.

"Understanding Synchronization User List Definitions" on page 311 describes the components of an SUL definition:

TABLE D–1 SUL Definition Components

Component	Definition	Applicable		
		Sun	AD	NT
Base DN	Defines the parent LDAP node of all users to be synchronized. A Synchronization User List base DN includes all users in that DN — unless the users are excluded by the Synchronization User List's filter or the user's DN is matched in a more specific Synchronization User List. For example, `ou=sales,dc=example,dc=com`.	Yes	Yes	No
Filter	Defines an LDAP-like filter used to include or exclude users from a Synchronization User List. The filter can include the &, |, !, =, and * operators. The \>= and <= operators are not supported. All comparisons are done using case-insensitive string comparisons. For example, `(& (employeeType=manager)(st=CA))` will include managers in California only.	Yes	Yes	Yes
Creation Expression	Defines the parent DN and naming attribute of newly created users (applicable only when you enable creates). The creation expression must include the base DN of the Synchronization User List. For example, `cn=%cn%,ou=sales,dc=example,dc=com`. (Where the %cn% token is replaced with a value from the user entry being created.)	Yes	Yes	No

---

Note – To synchronize users in a Sun Java System Directory Server with multiple Active Directory domains, you must define at least one SUL for each Active Directory domain.

When Group Synchronization is enabled, the following are important:

1. The creation expression supported at Active Directory is cn=%cn%.

2. The creation expression must contain valid attribute names belonging to the group objectclass since the creation expression is common to both user as well as group.

   For example:

   The attribute sn is not part of the `groupofuniquenames` objectclass at the Directory Server. Hence the following creation expression would be invalid for a group object. (Though it would work fine for user.)

   `cn=%cn%.%sn%`

3. The attribute used in the creation expression must be provided with a value for every user/group entry created. If the value is not provided then the user/group object will not synchronize and an appropriate message will be logged in the central log.

When you define multiple SULs, Identity Synchronization for Windows determines membership in an SUL by iteratively matching each SUL definition. The program examines the SUL definitions with more-specific base DNs first. For example, the program tests a match against `ou=sales,dc=example,dc=com` before testing `dc=example,dc=com`.

If two SUL definitions have the same base DN and different filters, then Identity Synchronization for Windows cannot determine automatically which filter should be tested first, so you must use the Resolve Domain Overlap feature to order the two SUL definitions. If a user matches the base DN of an SUL definition but does not match any filters for that base DN, then the program will exclude that user from synchronization — even if that user matches the filter for a less-specific base DN.

Configuring Multiple Windows Domains

To support synchronizing multiple Windows domains to the same Directory Server container (such as `ou=people,dc=example,dc=com`), Identity Synchronization for Windows uses "synthetic" Windows attributes that contain domain information.

- For Active Directory domains, Identity Synchronization for Windows sets the `activedirectorydomainname` attribute to the Active Directory domain name (such as *east.example.com*) before synchronizing the entry to the Directory Server.

- For Windows NT domains, Identity Synchronization for Windows sets the `user_nt_domain_name` attribute to the Windows NT domain name (such as *NTEXAMPLE*) before synchronizing the entry to the Directory Server.

While these attributes do not actually appear in the Windows user entries, they are available for synchronization in the Identity Synchronization for Windows Console and can be mapped to a Directory Server user attribute. Once Identity Synchronization for Windows maps the domain attributes, they will be set in the Directory Server entries during synchronization and can be used in Synchronization User List (SUL) filters.

The following example illustrates how Identity Synchronization for Windows uses these attributes. This example assumes that three Windows domains (two Active Directory domains and one Windows NT domain) will be synchronized with a single Directory Server instance.

▼ To Configure Multiple Windows Domains

1   **Users in the Active Directory east.example.com domain will be synchronized to the Directory Server in** `ou=people,dc=example,dc=com`**.**

2   **Users in the Active Directory west.example.com domain will be synchronized to the Directory Server in** `ou=people,dc=example,dc=com`**.**

3   **Users in the Windows NT NTEXAMPLE domain will be synchronized to the Directory Server in** `ou=people,dc=example,dc=com`**.**

When you create or modify a Directory Server user, the program uses the SUL filters to determine in which Windows domain to synchronize the user (because each Directory Server SUL has the same base DN, `ou=people,dc=example,dc=com`). The `activedirectorydomainname` and `user_nt_domain_name` attributes make constructing these filters easy.

To construct a filter from the Attributes tab on the Console:

4   **Map the Directory Server** `destinationindicator` **attribute to the Active Directory** `activedirectorydomainname` **attribute and to the Windows NT** `user_nt_domain_name` **attribute.**

5   **Configure one SUL for each Windows domain as follows:**

```
EAST_SUL

Sun Java System Directory Server definition
        Base DN:    ou=people,dc=example,dc=com
        Filter:     destinationindicator=east.example.com
        Creation Expression:  cn=%cn%,ou=people,dc=example,dc=com

Active Directory definition (east.example.com)
        Base DN:    cn=users,dc=east,dc=example,dc=com
        Filter:     <none\>
        Creation Expression:  cn=%cn%,cn=users,dc=east,dc=example,dc=com
```

```
WEST_SUL

Sun Java System Directory Server definition
        Base DN:ou=people,dc=example,dc=com
        Filter:  destinationindicator=west.example.com
        Creation Expression: cn=%cn%,ou=people,dc=example,dc=com

Active Directory definition (west.example.com)
        Base DN:  cn=users,dc=west,dc=example,dc=com
        Filter:<none\>
        Creation Expression:  cn=%cn%,cn=users,dc=west,dc=example,dc=com

NT_SUL

Sun Java System Directory Server definition
        Base DN:  ou=people,dc=example,dc=com
        Filter:  destinationindicator=NTEXAMPLE
        Creation Expression:  cn=%cn%,
        ou=people,dc=example,dc=com

Windows NT definition (NTEXAMPLE)
        Base DN:  NA
        Filter:  <none\>
        Creation Expression:  NA
```

Notice that each Directory Server SUL definition has the same base DN and creation expression, but the filters indicate the domain of the corresponding Windows user entry.

To further illustrate how these settings allow Directory Server user entries to synchronize with separate Windows domains, consider this test case:

6 **Create** cn=Jane Test,cn=users,dc=east,dc=example,dc=com **in the Active Directory east.example.com domain.**

7 **Identity Synchronization for Windows creates the user entry** cn=Jane Test,ou=people,dc=example,dc=com**in the Directory Server with** destinationindicator=east.example.com**.**

8 **Modify the**cn=Jane Test,ou=people,dc=example,dc=com **entry in the Directory Server.**

9 **Because Jane Test's** destinationindicator **attribute is east.example.com, her entry will match the** EAST_SUL **Synchronization User List filter, and the modification will be synchronized to the east.example.com Active Directory domain.**

   This example assumes that Identity Synchronization for Windows is synchronizing user creations from Windows to the Directory Server. If this is not the case, you can run the idsync resync command to set the destinationindicator attribute.

> **Note** – When you use `idsync resync -f` in a deployment with multiple SULs, you probably will have to set the `allowLinkingOutOfScope` option to `true` in the linking configuration file. See Appendix B

The example uses an existing attribute in `inetorgperson`, `destinationIndicator`, which might be used for other purposes. If this attribute is already in use or a you select a different objectclass, you must map some attribute in the user's Directory Server entry to the `user_nt_domain_name` and/or the `activedirectorydomainname` attribute(s). The Directory Server attribute you choose to hold this value must be in the objectclass you are using for the rest of the attribute mapping configuration.

If there are no unused attributes to hold this domain information, you must create a new objectclass to include a new domain attribute and all other attributes you will be using with Identity Synchronization for Windows.

E

Identity Synchronization for Windows Installation Notes for Replicated Environments

Identity Synchronization for Windows 6.0 supports synchronizing users in a single replicated suffix.

Note – This appendix summarizes procedures used to configure and secure a multimaster replication (MMR) deployment. The information is taken directly from the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide* — and is not Identity Synchronization for Windows - specific.

Designing and implementing an MMR deployment is *complex*. Refer to the *Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide* to plan your deployment and the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide* to implement the deployment.

This appendix is organized into the following sections:

Configuring Replication

Note – In multimaster replication (MMR) environments, Identity Synchronization for Windows allows you to specify a preferred and secondary master servers for any given Sun directory source.

Directory Server 6.0 now supports n-way MMR (where you can change the replicated database at any of the 'n' masters configured). When you install the plug-in at the preferred master, you must select the *Other* host type and enter Directory Server instance's parameters manually during plug-in installation.

The following steps assume you are replicating a single suffix. If you are replicating more than one suffix, you may configure them in parallel on each server. In other words, you may repeat each step to configure replication on multiple suffixes.

▼ To Configure any Replication Topology

1　Define a replication manager entry on all servers except single masters (or use the default replication manager on all servers.)

2　On all servers containing a dedicated consumer replica:

　　a.　Create an empty suffix for the consumer replica.

　　b.　Enable the consumer replica on the suffix through the replication wizard.

　　c.　Optionally, configure the advanced replica settings.

3　On all servers containing a hub replica, if applicable:

　　a.　Create an empty suffix for the hub replica.

　　b.　Enable the hub replica on the suffix through the replication wizard.

　　c.　Optionally, configure the advanced replica settings.

4　On all servers containing a master replica:

　　a.　Choose or create a suffix on one of the masters that will be the master replica.

　　b.　Enable the master replica on the suffix through the replication wizard.

    c.   **Optionally, configure the advanced replica settings.**

5   **Configure the replication agreements on all supplier replicas, in the following order:**

    a.   **Between masters in a multimaster set.**

    b.   **Between masters and their dedicated consumers.**

    c.   **Between masters and hub replicas.**

       Optionally, you can configure fractional replication at this stage.

6   **Configure replication agreements between the hub replicas and their consumers.**

7   **For multimaster replication, initialize all masters from the same master replica containing the original copy of the data. Initialize the hub and consumer replicas.**

Configuring Replication Over SSL

Note – In this procedure, all references are chapters in the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*.

▼ To Configure Directory Servers Involved in Replication so that all Replication Operations Occur Over an SSL Connection

1   **Configure both the supplier and consumer servers to use SSL.**

Refer to Chapter 11, "Managing Authentication and Encryption" for details.

Note –

- Replication over SSL will fail if the supplier server certificate is an SSL server-only certificate that cannot act as a client during an SSL handshake.
- Replication over SSL is currently unsupported with self-signed certificates.

2   **If replication is not configured for the suffix on the consumer server, enable it as described in Chapter 8, "Enabling a Consumer Replica."**

3   **Follow the procedure in Chapter 8, "Advanced Consumer Configuration," to define the DN of the certificate entry on the consumer as another replication manager.**

4   If replication is not configured for the suffix on the supplier server, enable it as described in Chapter 8, "Enabling a Hub Replica" or "Enabling a Master Replica."

5   On the supplier server, create a new replication agreement to send updates to the consumer on the secure SSL port. Follow the procedure in Chapter 8, "Creating Replication Agreements," for detailed instructions. Specify a secure port on the consumer server and select the SSL option of either using a password or a certificate. Enter a DN for the SSL option that you chose, either a replication manager or a certificate.

After you finish configuring the replication agreement, the supplier will send all replication update messages to the consumer over SSL and will use certificates if you chose that option. Customer initialization will also use a secure connection if performed through the console using an agreement configure for SSL.

Configuring Identity Synchronization for Windows in an MMR Environment

▼ To Configure Identity Synchronization for Windows in an MMR Environment

1   From the Identity Synchronization for Windows Console, specify a preferred master and secondary master servers for the suffix to be synchronized. (Review "Creating a Sun Java System Directory Source" on page 146.)

You do not have to provide information about other Directory Servers in your topology.

2   Prepare the preferred master and secondary master servers from the Console or using the `idsync prepds` command line utility. (Review "Preparing Sun Directory Source" on page 153

If you use the command line utility, you should prepare both servers in a single invocation by specifying arguments for both the preferred and secondary servers.

3   Install the Directory Server Connector for the suffix replicated between these directories. (Review "Installing the Directory Server Connector" on page 200.)

4   Configure the Directory Server Plug-in on the preferred master, the secondary masters, and every other Directory Server instance that manages users in the replicated suffix(Review "Using dspluginconfig" on page 301.)

Index