Sun Java™ System

Identity Synchronization for Windows 1 Installation and Configuration Guide

2004Q3

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817-6199-05

Contents

List of Figures

List of Tables

Preface

Sun Java™ System Identity Synchronization for Windows 1 2004Q3 (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 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 enhancements in this release of Identity Synchronization for Windows, please see the online release notes at:

`http://docs.sun.com/db/doc`

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

This Preface contains the following information:

Who Should Use This Book

This *Installation and Configuration Guide* is intended for use by administrators, systems engineers, and professional services engineers who will install and configure Identity Synchronization for Windows to establish bidirectional password and user attribute synchronization between Sun Java™ System Directory Server and Windows Active Directory / NT machines.

You should already be familiar with

- Configuring and operating Directory Server and Windows Active Directory / NT
- Lightweight Directory Access Protocol (LDAP)
- Java technology
- Extensible Markup Language (XML)
- Basic concepts of public-key cryptography and Secure Sockets Layer (SSL) protocol
- Basic concepts of intranet, extranet, and the Internet security and the role of digital certificates in an enterprise

Before You Read This Book

The *Sun Java System Identity Synchronization for Windows 1 2004Q3 Release Notes* contain the latest information about the product — including information that may supersede instructions provided in this book. Be sure you read these Release Notes before attempting any procedures described in this book.

Because Sun Java System Directory Server is used as the data store in an Identity Synchronization for Windows deployment, you should be familiar with the documentation provided with that product. Directory Server documentation can be accessed online at `http://docs.sun.com/coll/DirectoryServer_04q2`.

How This Book Is Organized

The *Sun Java System Identity Synchronization for Windows 1 2004Q3 Installation and Configuration Guide* is organized into the following chapters:

- Chapter 1, “Understanding the Product”: Explains some basic concepts related to Identity Synchronization for Windows; such as product features, system components, command line utilities, system component distribution, and deployment examples.
- Chapter 2, “Preparing for Installation”: Describes the installation and configuration processes, and provides information you may find helpful as you prepare to install the product.
- Chapter 3, “Installing Core”: Explains how to use the Identity Synchronization for Windows installation program and how to install the Identity Synchronization for Windows Core component.
- Chapter 4, “Configuring Core Resources”: Explains how to add and configure Core resources using the Console.
- Chapter 5, “Installing Connectors and Directory Server Plugins”: Provides instructions for installing the Identity Synchronization for Windows Connectors and Directory Server Plugins.
- Chapter 6, “Synchronizing Existing Users”: Explains how to link and resynchronize existing users for new Identity Synchronization for Windows installations.
- Chapter 7, “Migrating to Identity Synchronization for Windows 1 2004Q3”: Explains how to migrate your system from Sun Java System Identity Synchronization for Windows version 1.0 to version 1 2004Q3.
- Chapter 8, “Removing the Software”: Explains how to remove Identity Synchronization for Windows, including how to prepare for the uninstallation and how to uninstall the Console manually.
- Chapter 9, “Troubleshooting”: Provides information you can use to troubleshoot your Identity Synchronization for Windows installation.
- Chapter 10, “Understanding Audit and Error Files”: Provides information about audit and error logging, including how to set logging levels, viewing and understanding your log files and directory source status.
- Chapter 11, “Configuring Security”: Describes how to configure a secure system. Information provided includes hardening security, securing replicated configurations, enabling SSL, and adding Active Directory CA certificates to certificate databases.

- Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities”: Explains how to use the Identity Synchronization for Windows command line utilities to perform different tasks.
- Appendix B, “LinkUsers XML Document Sample”: Provides a sample Linkusers XML document (`linkusers-simple.cfg`) that you can use to customize to your environment.
- Appendix C, “Running Services as Non-Root on Solaris”: Explains how to run Identity Synchronization for Windows services as a non-root user.
- Appendix D, “Defining and Configuring Synchronization User Lists”: Provides information about Synchronization User List definitions and multiple domain configurations.
- Appendix E, “Installation Notes for Replicated Environments”: Provides a brief overview of the steps required to configure and secure a multimaster replication (MMR) deployment.

Conventions Used in This Book

The tables in this section describe the conventions used in this book. The information is organized as follows:

Typographic Conventions

The following table describes the typographic conventions used in this book.

Table 1 Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, on-screen computer output, sample code.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
AaBbCc123 (Monospace bold)	What you type, when contrasted with on-screen computer output.	`% **su**` `Password:`
AaBbCc123 (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. The file is located in the *install-dir*`/bin` directory.

Symbols

The following table describes the symbol conventions used in this book.

Table 2 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional 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.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	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.

Mnemonics

Identity Synchronization for Windows uses *mnemonics* (underlined letters) throughout the user interface to give you quicker options for performing certain tasks. You simply type the unlined letter to perform the task. Mnemonics are not case sensitive. To access them press the Alt key simultaneously.

For example, in some dialog boxes, you can type a capital “C” or “Alt-c” to cancel the dialog box or type capital “H” or “Alt-h,” to opent an online help dialog box.

Default Paths and File Names

The following table describes the default paths and file names used in this book.

Table 3 Default Paths and File Names

Term	Description
<*serverroot*>	Represents represents the parent directory of the Identity Synchronization for Windows installation location
`isw-`<*hostname*>	Represents the Identity Synchronization for Windows instance directory
<*current-working-directory*>`/cert8.db`	Represents the default path and file name of the client’s certificate database
<*installation_root*>`/isw-`<*machine_name*>`/logs/central/`	Represents the default path to the Identity Synchronization for Windows central logs
<*installation_root*>`/isw-`<*machine_name*>`/logs/`	Represents the default path to the Identity Synchronization for Windows local logs (for the System Manager, each connector, and the Central Logger)
`/usr/sfw/bin`	On Solaris, certutil is installed in this directory location by default.

Related Documentation

The http://docs.sun.comweb site enables you to access Sun technical documentation online. You can browse the archive or search for a specific book title or subject.

Books in This Documentation Set

The following table summarizes the books included in the Identity Synchronization for Windows documentation set.

Table 4 Books in This Documentation Set

Book Title	Description
Sun Java System Identity Synchronization for Windows 1 2004Q3 Installation and Configuration Guide (http://docs.sun.com/doc/817-6199)	Describes how to install and configure Identity Synchronization for Windows for use in a production environment.
Sun Java System Identity Synchronization for Windows 1 2004Q3 Deployment Planning Guide (http://docs.sun.com/doc/817-6200)	Provides general guidelines and best practices for the planning and deploying Identity Synchronization for Windows.
Sun Java System Identity Synchronization for Windows 1 2004Q3 Release Notes (http://docs.sun.com/doc/817-6202)	Available after the product is released. Contains last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Other Documentation

Because you will be working ith Directory Server and Sun Java™ System Message Queue, you may need to refer to their product documentation. You can access the documentation from the following locations:

- Sun Java System Directory Server documentation
  http://docs.sun.com/coll/DirectoryServer_04q2
- Sun Java System Message Queue documentation
  http://docs.sun.com/db/prod/2296#hic

For information about the basic concepts of public-key cryptography; Secure Sockets Layer (SSL) protocol; intranet, extranet, and the Internet security; and the role of digital certificates in an enterprise, read the security-related appendixes in the *Managing Servers with iPlanet Console 5.0* manual.

For information about Windows 2003 Server and Windows Password Policies, read the following Microsoft publications:

- *Using Secedit.exe to Force Group Policy to Be Applied Again - Windows 2000 Servers Microsoft KB #227448*
- *A Description of the Group Policy Update Utility - Windows 2003 Servers Microsoft KB #298444*
- *Microsoft Knowledge Base Article 232690*

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

- Developer Information
  http://developers.sun.com/prodtech/index.html
- Download Center
  http://wwws.sun.com/software/download/
- Product Data Sheets
  http://www.sun.com/software/
- Product Documentation Online
  http://docs.sun.com
- Product Support and Status
  http://www.sun.com/service/support/software/
- Professional Services
  http://www.sun.com/service/sunps/sunone/index.html
- Sun Enterprise Services, Solaris Patches, and Support
  http://sunsolve.sun.com
- Support and Training
  http://www.sun.com/supportraining/

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to:

http://www.sun.com/service/contacting

Related Third-Party Web Site References

The following third-party web sites are referenced in this publication:

- For information about password policies for Windows 2003:

  http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/deployguide/en-us/dsscc_aut_xbby.asp

- For information about applying or modifying passwords and group policies in Windows 2003:

  http://www.microsoft.com/resources/documentation/windowsserv/2003/standard/proddocs/en-us/password_grouppolicy.asp

- For information about the Microsoft Certificate Services Enterprise Root certificate authority:

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

- For information about configuring LDAP over SSL:

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

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to http://docs.sun.com and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the book or at the top of the document.

For example, the title of this books is *Sun Java System Identity Synchronization for Windows 1 2004Q3 Installation and Configuration Guide*, and the part number is 817-6199.

Part I

Installation and Configuration

Chapter 1

Understanding the Product

Identity Synchronization for Windows provides bidirectional password and user attribute synchronization between the Sun Java™ System Directory Server 5 2004Q2 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 1 2004Q3, 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 Sun Java System directory sources and Windows Active Directory and Windows NT directory sources.

  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 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 directory sources (not available for Windows NT).

- **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.

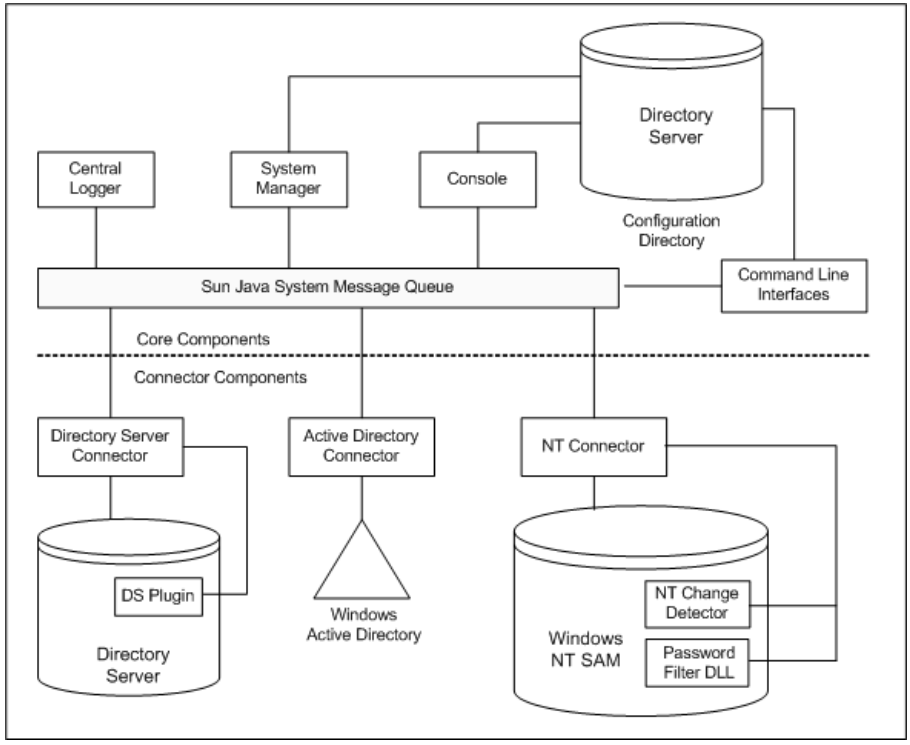
If you are using Identity Synchronization for Windows to synchronize between Directory Server and Active Directory, you will not be required to install any components in the Windows operating environment.

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

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 (see Figure 1-1).

Figure 1-1 System Components



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

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 or a Windows service. (For information about starting and stopping services, see “Starting and Stopping Services” on page 188.)

Core

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

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

- “Configuration Directory” on page 31
- “Console” on page 31
- “Command Line Utilities” on page 32
- “System Manager” on page 32
- “Central Logger” on page 33

NOTE The Watchdog is installed where you install Core, and it is responsible for launching and monitoring the central logger and system manager.

For more information, see “Watchdog Process” on page 30.

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 Plugin
- Connector status
- Synchronization settings that describe the direction of user creations, user 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

- 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
- Import an exported Identity Synchronization for Windows version 1.0 configuration XML document
- 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/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
- Start synchronization
- Stop synchronization

For a detailed description of the product's command line utilities and how to use them, see Appendix A, "Using the Identity Synchronization for Windows Command Line Utilities."

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 and connectors. For more information, see “Watchdog Process” on page 30.

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 installed with the directory being synchronized and communicate with the corresponding connector over an encrypted connection.

- “Directory Server Plugin” on page 35
- “Windows NT Connector Subcomponents” on page 35

NOTE Active Directory Connectors do not require subcomponents.

Directory Server Plugin

The Directory Server Plugin is a subcomponent of the Directory Server Connector. You install the Directory Server Plugin in each Directory Server being synchronized.

This Plugin

- 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 43)

NOTE The Directory Server Plugin is functional in four-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:

- “Core” on page 37
- “Directory Server Connector and Plugin” on page 37
- “Active Directory Connector” on page 38
- “Windows NT Connector and Subcomponents” on page 39

When you understand the basic concepts described in this section and in the Deployment Scenario example (on page 48), 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

You first install all Core components only once in any of the supported operating system's directory servers. The Administration Server must reside on the same machine as Core. You must install Message Queue 3.5 SP1 Enterprise Edition prior to installing Core.

Directory Server Connector and Plugin

You can install Directory Server Connectors on any of the supported operating systems (listed in "Operating System Requirements" on page 54). 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 per each configured Directory Server source.

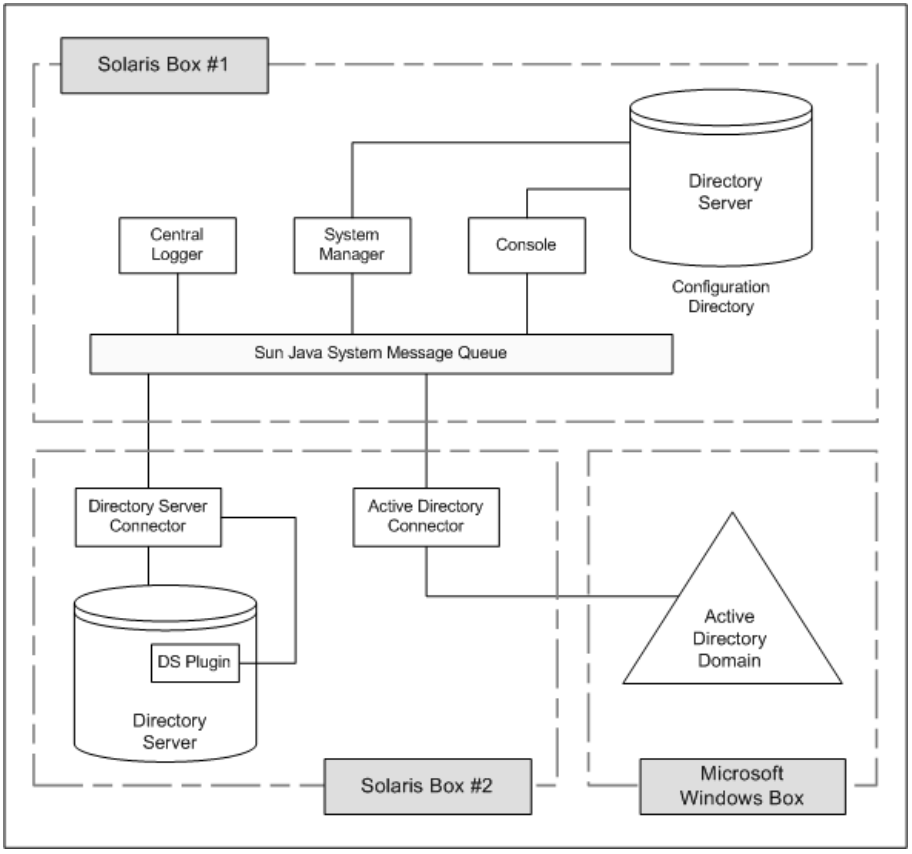
You must install the Directory Server Plugin on every host where a Directory Server to be synchronized resides.

NOTE A single Directory Server Connector is installed for each Directory Server source. However, Directory server Plugins should be installed 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 (see Figure 1-2). 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.

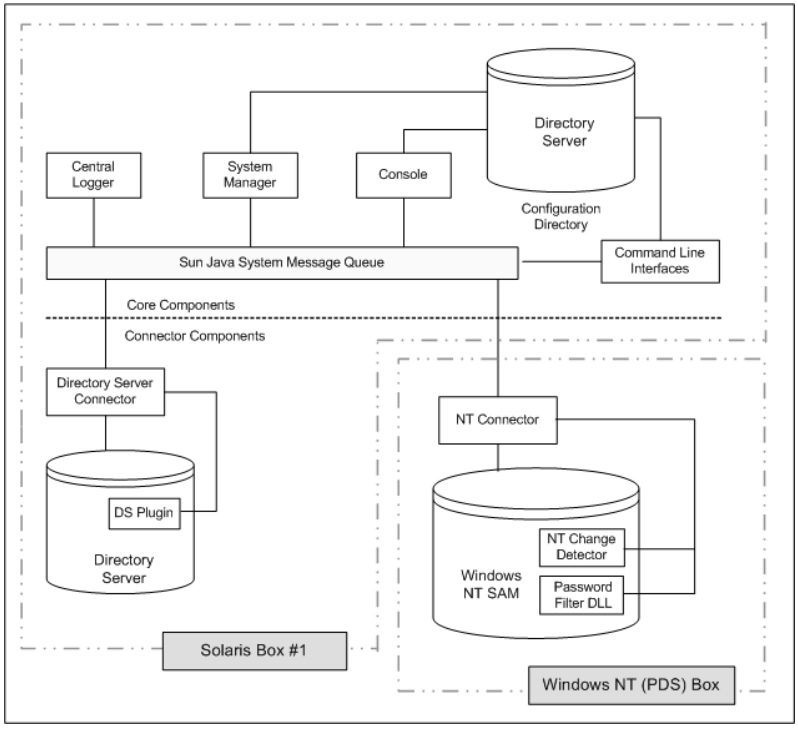
Figure 1-2 Directory Server and Active Directory Component Distribution



Windows NT Connector and Subcomponents

To synchronize with Windows NT SAM Registries (see Figure 1-3) 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 1-3 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.

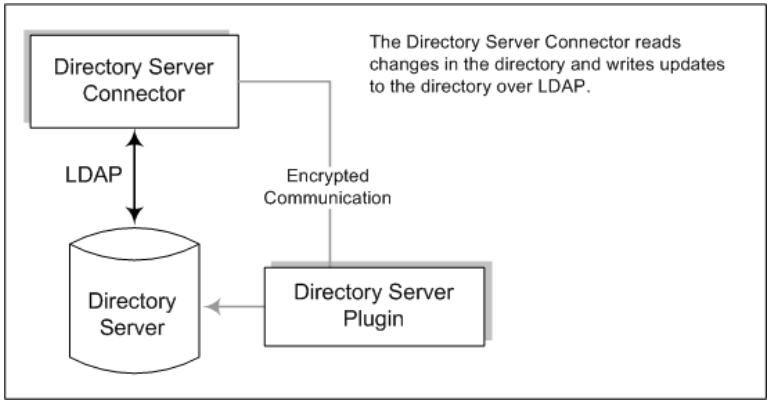
The information is organized as follows:

How Directory Server Connectors Detect Changes

The Directory Server Connector examines the Directory Server Retro-Changelog over LDAP to detect user entry and password change events. The Directory Server Plugin helps the Connector:

- 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 43.)

Figure 1-4 How Directory Server Connectors Detect Changes



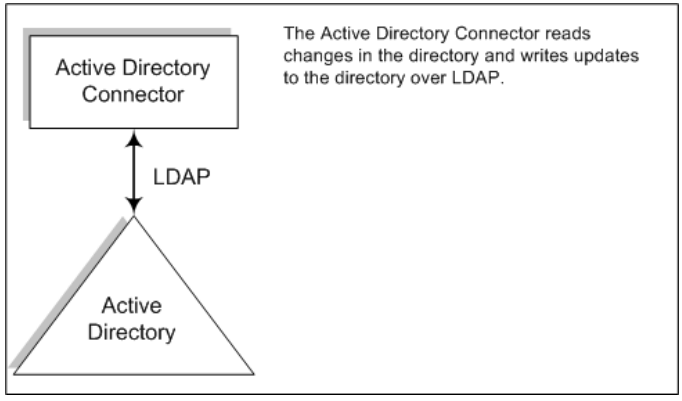
How Active Directory Connectors Detect Changes

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 elsewhere (such as a Solaris computer) and detect or make changes remotely over LDAP.

Figure 1-5 How Active Directory Connectors Detect Changes

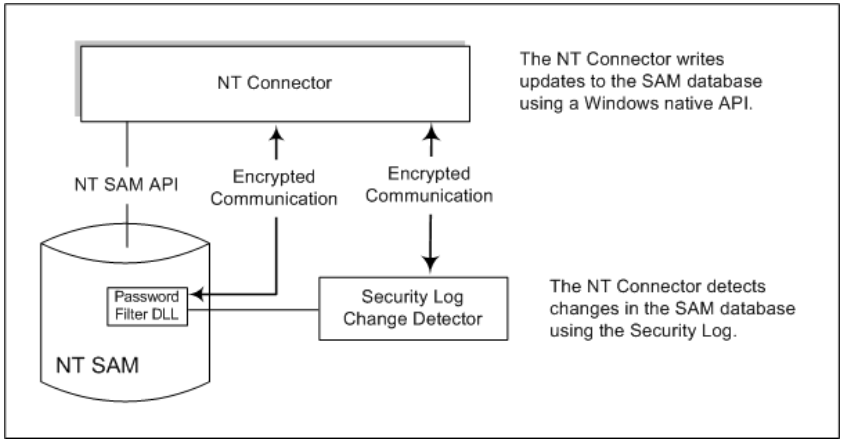


How Windows NT Connectors Detect Changes

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 Figure 1-3) 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 1-6 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 log messages from the that machine. To verify whether audit logging is enabled on your Windows NT machine, see “Enabling Auditing on a Windows NT Machine” on page 283.

For a description of the Change Detector and the Password Filter DLL subcomponents, review “Windows NT Connector Subcomponents” on page 35.

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 43
- "Using On-Demand Password Synchronization to Obtain Clear-Text Passwords" on page 43

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.

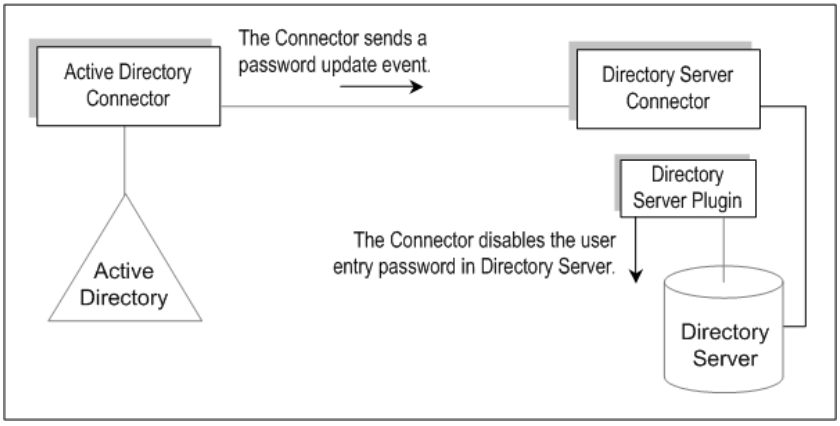
On-demand password synchronization also allows you to synchronize passwords on Active Directory without the need of a password filter DLL.

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 1-7 On-Demand Password Synchronization — Part I



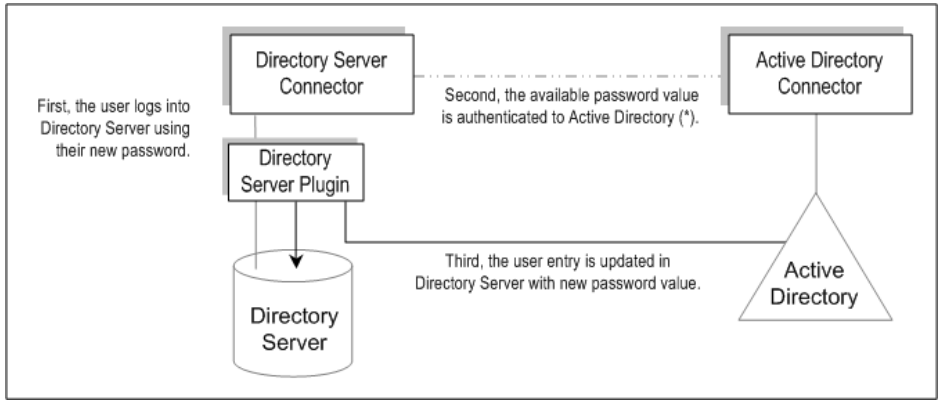
3. The Directory Server Connector receives the password change message from Message Queue (over SSL).
4. The Directory Server Connector sets the user entry's `dspswvalidate` attribute to `true`., which invalidates the old password and alerts the Directory Server Plugin of the password change.
5. When 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 Plugin detects that the password value in the Directory Server entry is invalid.

6. The Directory Server Plugin searches for the corresponding user in Active Directory. When the Plugin finds the user, the Plugin 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 Plugin set the password and removed the invalid password flag from the user entry on Directory Server.

Figure 1-8 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 multimaster replication (MMR) Directory Server environment, external changes to Windows users can be synchronized to the preferred or secondary Directory Server.

  If the preferred Directory Server is unavailable, then the Directory Server Connector will apply changes to the secondary server. Identity Synchronization for Windows will not detect and propagate external changes made on any Directory Server master until the preferred master is available.

- While the Active Directory Connector can communicate with a single Active Directory domain controller only, the Directory Server Plugin 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 Plugin 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.

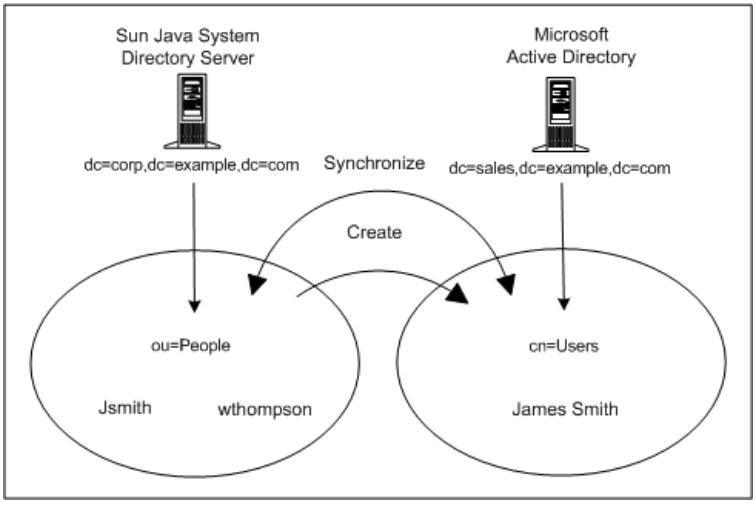
The deployment scenario consists of two systems:

- 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.

Figure 1-9 illustrates the synchronization requirements (node structures with associated attribute values) used for this deployment scenario.

Figure 1-9 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.

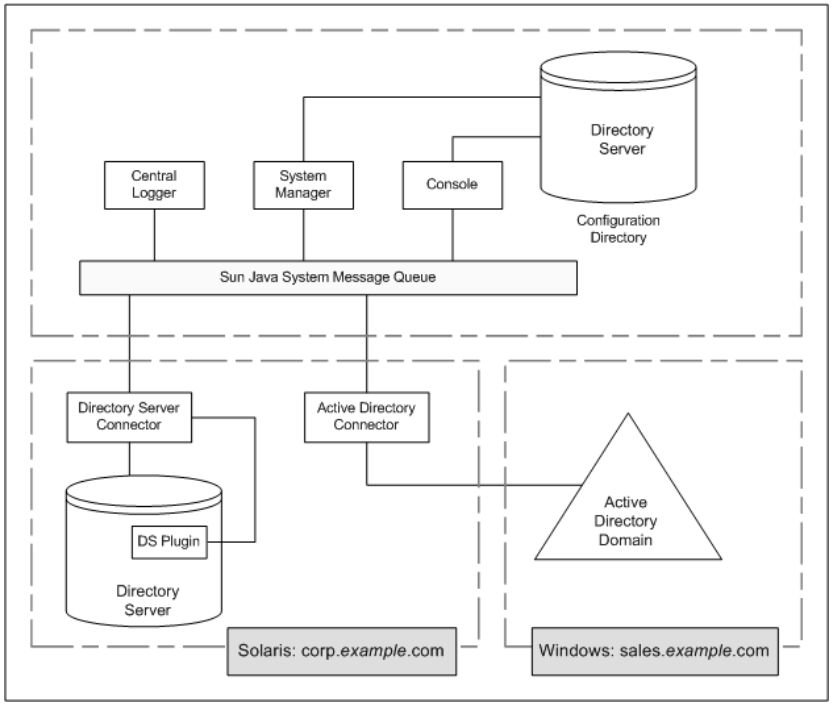
NOTE Identity Synchronization for Windows supports multiple synchronization sources of the same type (for example, you can have more than one Directory Server in a deployment or multiple Active Directory domains).

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 Sun to Windows, then user object creations will propagate from *all* Sun Directory Servers to *all* Active Directory domains and Windows NT domains configured in the installation.

Physical Deployment

Figure 1-10 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 1-10 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 Plugin
- 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.

Chapter 2

Preparing for Installation

Before installing Identity Synchronization for Windows 1 2004Q3 or before you migrate from version 1.0 to version 1 2004Q3, you should familiarize yourself with the installation and configuration process.

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 Requirements

This section describes the installation requirements for Identity Synchronization for Windows, which includes operating system versions, patches, and utilities for each platform.

Operating System Requirements

The following tables describe the operating system requirements for this release of Identity Synchronization for Windows:

Table 2-1 Solaris Requirements

Component	Solaris Requirement
Core Components	Solaris 8™ for UltraSPARC® (32-bit and 64-bit) Solaris 9™ SPARC® Platform Edition (32-bit and 64-bit) Solaris 9™ Operating System (x86 Platform Edition for Pentium II or later) IA-32
Connectors for Sun Java™ System Directory Server and for Windows Active Directory	Solaris 8 for UltraSPARC (32-bit and 64-bit) Solaris 9 for SPARC platforms (32-bit and 64-bit) Solaris 9 Operating System (x86 Platform Edition for Pentium II or later) IA-32
Plugin for Sun Java™ System Directory Server	Solaris 8 for UltraSPARC (32-bit and 64-bit) Solaris 9 for SPARC platforms (32-bit and 64-bit) Solaris 9 Operating System (x86 Platform Edition for Pentium II or later) IA-32

Table 2-2 Windows Requirements

Component	Windows Requirement
Core	Windows 2000 Server SP4 Windows 2000 Advanced Server SP4
	Windows 2003 Server Standard Edition (with latest security updates)
	Windows 2003 Server Enterprise Edition (with latest security updates)
Connectors for Sun Java™ System Directory Server and for Windows Active Directory	Windows 2000 Server SP4 Windows 2000 Advanced Server SP 4
	Windows 2003 Server Standard Edition (with latest security updates)
	Windows 2003 Server Enterprise Edition (with latest security updates)
Plugin for Sun Java™ System Directory Server	Windows 2000 Server SP4 Windows 2000 Advanced Server SP 4
	Windows 2003 Server Standard Edition (with latest security updates)
	Windows 2003 Server Enterprise Edition (with latest security updates)
NT Connectors and subcomponents	Windows Primary Domain Controller NT 4.0 Server SP 6A (for x86 only)

Hardware Requirements

Your hardware (all platforms) must meet the following minimum requirements to run Identity Synchronization for Windows:

- Approximately 400 MB of disk space for a minimal installation on Directory Server.
- A minimum of 512 MB of RAM for servers running any Identity Synchronization for Windows component. (more memory is preferred)

Sun Java System Software Requirements

Before you can install Identity Synchronization for Windows, you must install the following Sun Java System software components:

- Sun Java System Directory Server version 5 2004Q2 Patch 117907-02 (or higher)

  The patch fix enables delete functionality for Identity Synchronization for Windows 1 2004Q2 with Directory Server 5 2004Q2.

  - **For Solaris SPARC Package Format**: Patch Number 117907-02 or higher
  - **For Solaris SPARC Compressed Archive Installations**: Patch 5077789
  - **For Solaris x86 Package Format**: Patch Number 117908-02 or higher
  - **For Solaris x86 Compressed Archive Installations**: Patch 5077789
  - **For Windows Compressed Archive Installations**: Patch 5077789

  For additional details on these patches and how to apply them to your Directory Server environment, please see the `README.patch` file located in the Identity Synchronization for Windows download, available from:

  <download_root>`/patches/directory/README.patch`

  For the latest information about patches that may be required to install Directory Server 5 2004Q2 on Solaris, refer to the *Sun Java System Directory Server 5 2004Q2 Installation and Tuning Guide* and the *Sun Java System Directory Server 5 2004Q2 Release Notes*, which can be found at the following web site:

  `http://docs.sun.com/coll/DirectoryServer_04q2`

- Sun Java System Message Queue (formerly Sun ONE Message Queue) version 3.5 SP1 Enterprise Edition.

NOTE Identity Synchronization for Windows version 1.0 installed Message Queue for you, *but version 1 2004Q3 does not*.

To install the Identity Synchronization for Windows Core on an *existing* Sun Java System Message Queue installation, you must be using Message Queue version 3.5 SP1 Enterprise Edition. Trying to install Core on an improper version of Message Queue will fail.

The Identity Synchronization for Windows download bundle includes Message Queue. The software is available in the `/messagequeue` directory for each platform, as follows:

- **Solaris SPARC**: `/messagequeue/imq3_5-ent-solsparc.zip`
- **Solaris x86**: `/messagequeue/imq3_5-ent-soli386.zip`
- **Windows**: `/messagequeue/imq3_5-ent-win.exe`

- Java Runtime Environment

  A Java Runtime Environment (JRE) is not provided with this product.

  - You must install J2SE (or JRE) 1.4.2_04 (or later) to run the Identity Synchronization for Windows installer on Solaris or Windows.
  - You must install JRE 1.4.1_03 (or later) on Windows NT.

Installation Credentials

To install Identity Synchronization for Windows, you must provide credentials for the following:

- Configuration directory
- Directory Server being synchronized
- Active Directory (See “Installing Core” for more information.)

In addition, you must have the following privileges to install Identity Synchronization for Windows:

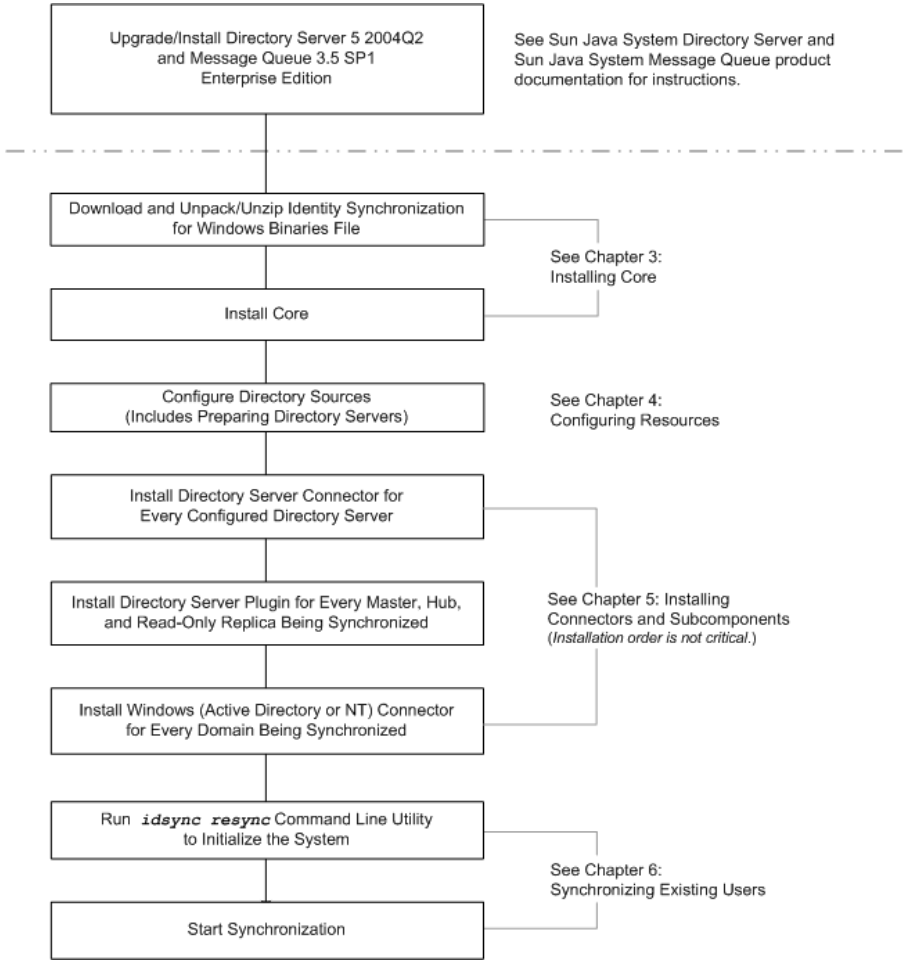
- **On Solaris systems**: You must install as *root*.
- **On Windows systems**: You must install as *Administrator*.

NOTE When you enter passwords using the text-based installer, the program automatically masks the passwords so they will not be echoed in the clear. The text-based installer is supported on Solaris systems only.

Installation Overview

Figure 2-1 illustrates the process for installing the product for a single-host deployment.

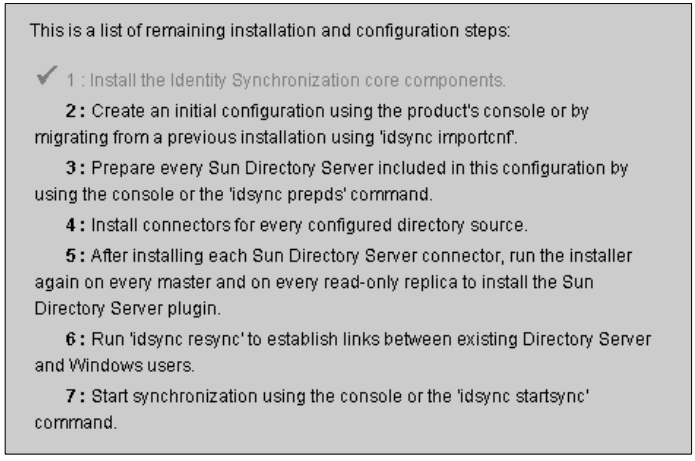
Figure 2-1 Installing in a Single-Host Deployment



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.

Figure 2-2 Identity Synchronization for Windows To Do List



As you go through the installation and configuration process, the program greys-out all completed steps in the list (as you can see in Figure 2-2).

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

- “Installing Core” on page 60
- “Configuring the Product” on page 60
- “Preparing the Directory Server” on page 60
- “Installing the Connectors and the Directory Server Plugin” on page 61
- “Synchronizing Existing Users” on page 62

NOTE Detailed installation and configuration instructions are provided later in this manual.

Installing Core

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

- **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 3, "Installing Core."

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 4, "Configuring Core Resources."

Preparing the Directory Server

Directory Server Connectors support the Sun Java System Directory Server 5 2004Q2.

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 the Directory Server" on page 115.

Installing the Connectors and the Directory Server Plugin

You can install any number of connectors and Directory Server Plugins, based on how many configured directories there are in your system.

NOTE The Console and the installation program both use a directory's *label* to associate a connector with the directory being synchronized. Table 2-3 describes Identity Synchronization for Windows's label-naming conventions.

Table 2-3 Label Naming Conventions

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

NOTE Instructions for installing and configuring Connectors and Directory Server Plugins are provided in Chapter 5, "Installing Connectors and Directory Server Plugins."

Synchronizing Existing Users

After installing the connectors, plugins, 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 a definition of *linking*, see “Linking Users” on page 182.)
- 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.

NOTE Instructions for synchronizing existing users in your deployment are provided in Chapter 6, “Synchronizing Existing Users.”

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 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 63
- "Configuration Directories and Global Catalogs" on page 64
- "Synchronization Settings" on page 64
- "Objectclasses" on page 64
- "Attributes and Attribute Mapping" on page 65
- "Synchronization User Lists" on page 67

NOTE Detailed configuration instructions are provided later in this manual.

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.

Configuration Directories and Global Catalogs

Identity Synchronization for Windows uses Sun Java System Directory Server configuration directories and Active Directory global catalogs as repositories in which to fetch Directory Server or Active Directory directory topology — as well as schema information for these directories.

Synchronization Settings

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

- From Sun to Windows
- From Windows to Sun
- 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 an Sun and between Active Directory and Sun.

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 4, "Configuring Core Resources."

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 4.

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 66.)

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=%B%`, 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 to synchronize by mapping the attribute names between the Sun and Windows systems. 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.

Migrating to Version 1 2004Q3

The procedures you use to migrate from Identity Synchronization for Windows version 1.0 (or version 1.0 SP1) are similar to the procedures you would use for a first-time 1 2004Q3 installation, with a few exceptions.

NOTE Migration procedures are provided in Chapter 7

Before you migrate to Identity Synchronization for Windows 1 2004Q3 you should be aware of the following:

- You must restore the Directory Server Connector state file and the Active Directory and NT Connector object cache files manually after installing the connectors. Be sure you have sufficient disk space (based on the size of the `/isw-home/persist` directories and subdirectories) on which to save a copy of each Active Directory and NT Connector object cache.
- You must uninstall all version 1.0 and 1.0 SP1 components.

  If the version 1 2004Q3 installer finds remnants of the version 1.0 system, it may cause problems with the Identity Synchronization for Windows schema installed into Directory Server and the actual Identity Synchronization for Windows binaries installed on the machine.

  NOTE For more information, see “What to Do if the 1.0 Uninstallation Fails” on page 212.

- You must install the Identity Synchronization for Windows 1 2004Q3 components on the same platform and hardware architecture as they were installed for 1.0.

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.

  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.

This rest of this section is organized as follows:

- “Enforcing Password Policies” on page 70: If you must enforce password policies on Windows or on Directory Server, read the information provided in this section to understand how password policies can affect synchronization results between Active Directory and Directory Server.
- “Example Password Policies” on page 75: This section provides password policy examples for several different scenarios.

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 5 2004Q2 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 When Identity 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:

- Table 2-4 describes how password policies affect synchronization.
- Table 2-5 describes how password policies affect 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 2-4 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 71 for more information.
	No	Yes	**Yes**	Yes	See “Important Notes” on page 70 for more information.
	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 71 for more information.
Directory Server	Yes	Yes	Yes	Yes	
	Yes	No	Yes	No	
	No	Yes	No	No	
	No	No	No	No	

Table 2-5 How Password Policies Affect Resynchronization Behavior

Scenario			Result
Resync Command	**User Meets Password Policy In**		
	Directory Server	**Active Directory**	
`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 72 for more information.
	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 72 for more information.
`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 72 for more information.
	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 70 and “Creating Accounts Without Passwords” on page 72 for 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 72 for more information.
	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 72 for more information.

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/all/deployguide/en-us/dsscc_aut_xbby.asp

For more information about password policies for Directory Server 5 2004Q2, see

http://docs.sun.com/db/coll/DirectoryServer_04q2

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 you must 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 302.

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
- Core Configuration
- Connector and Directory Server Plugin Installation
- Using the Command Line Utilities

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.

  NOTE Identity Synchronization for Windows does not support the configuration directory being installed as *localhost*.

- **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 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.

Core Configuration

You must provide the following information when you configure Core:

- **Sun Java System Directory schema server**: 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 Sun to Windows
  - From Windows to Sun
  - 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 and Directory Server Plugin Installation

You must provide the following information when you install the connectors and the Directory Server Plugin:

- **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 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 plugin.

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 Plugin, 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 following utilities:

- You use the `idsync` script with the following subcommands to execute the Identity Synchronization for Windows command line utility:
  - `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
  - `startsync:` Starts synchronization
  - `stopsync:` Stops synchronization

NOTE See Appendix A for detailed information about these utilities.

- You the following utilities to migrate from Identity Synchronization for Windows 1.0 or 1.0 SP1 to Identity Synchronization for Windows 1 2004Q3:
  - `forcepwchg`: Requires a password change for users who changed their passwords during the Identity Synchronization for Windows version 1.0 to version 1 2004Q3 migration process
  - `importcnf:` Imports an exported version 1.0 configuration XML document

NOTE See Chapter 7 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 2-6 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	

Table 2-7 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	

Table 2-7 Core Configuration Checklist *(Continued)*

Required Information	Entry
Synchronization User Lists	
Windows source filter creation expression	
Sun Java System source filter creation expression	

Table 2-8 Connector and Directory Server Plugin 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 Plugin	

Table 2-9 Linking Users Checklist

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

Table 2-10 Resynchronization Checklist

Required Information	Entry
Synchronization User List selection	
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?	

Chapter 3

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 2, "Preparing for Installation." This chapter 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 1.4.2_04 (or later) to run the Identity Synchronization for Windows installation program on your Solaris or Windows 2000/2003 systems.
- **On Windows systems only**: You must close any open Service Control Panel windows before starting Core installation, or the installation will fail.

- If you still have Identity Synchronization for Windows version 1.0 (or 1.0 SP1) installed on your machine, read Chapter 7, "Migrating to Identity Synchronization for Windows 1 2004Q3."

NOTE The Identity Synchronization for Windows 1.0 uninstall program will remove the `SUNWjss` package if it is not registered for use by another application (other than Identity Synchronization for Windows 1.0). In particular, this situation may occur on Solaris machines if you installed a zip version of Directory Server 5.2.2, where the uninstall program removes the `jss3.jar` file from `/usr/share/lib/mps/secv1`.

If you encounter this situation as you migrate to Identity Synchronization for Windows 11 2004Q3, the installer will report that a required file is missing, and log the file name to the installation log. When this happens, you must re-install the required patches (see "Sun Java System Software Requirements" on page 56) and restart the installation process.

- Identity Synchronization for Windows version 1.0 installed Message Queue for you, *but version 1 2004Q3 does not*. You should have already installed Message Queue 3.5 SP1 Enterprise Edition.

  On Solaris 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:

1. Log in as root.
2. Create a new directory by typing **# mkdir isw12004Q3**, and then change (**cd**) to that directory.
3. If you have not already done so, download the product binaries file (`isw-12004Q3.sparc-sun-solaris.tar.gz`) to the installation directory.
4. Use the following command to unpack the product binaries file:

   ```
   # gunzip -dc isw-12004Q3.sparc-sun-solaris.tar.gz | tar -xvof -
   ```

5. From the `isw12004Q3` directory, 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 Solaris x86

Use the following steps to prepare and run the Identity Synchronization for Windows installation program on a Solaris x86 operating system:

1. Log in as root.
2. Create a new directory by typing **# mkdir isw12004Q3**, and then change (**cd**) to that directory.
3. If you have not already done so, download the product binaries file (`isw-12004Q3.x86-sun-solaris.tar.gz`) to the installation directory.
4. Use the following command to unpack the product binaries file:

   ```
   # gunzip -dc isw-12004Q3.x86-sun-solaris.tar.gz | tar -xvof -
   ```

5. From the `isw12004Q3` directory, 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

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

1. Log in as an Administrator.
2. Create a new directory by typing **`# mkdir isw12004Q3`**
3. Change (**`cd`**) to the `isw12004Q3` directory.
4. If you have not already done so, download the product binaries file (`isw-12004Q3-windows.zip`) to the installation directory.
5. Unzip the `isw-12004Q3-windows.zip` file to an empty directory.
6. From the `isw12004Q3` directory, **`cd`** to the `installer` directory and then type **`setup.exe`** to execute the installation program.

   The Identity Synchronization for Windows installation wizard is displayed.

NOTE Because you are installing Core in the Administration Server root, the Identity Synchronization for Windows wizard will detect most of the information required for installation (such as directory paths and names) and will 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.

Installing Core

This section explains the process for installing the Identity Synchronization for Windows Core on both Solaris 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.

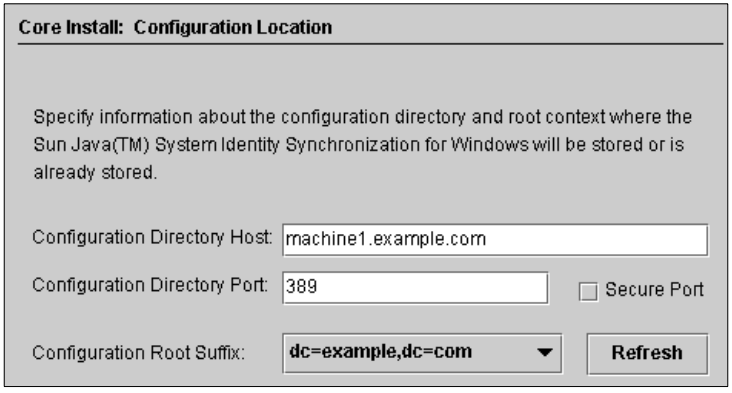
  NOTE You must install the program as root, but after installation you can configure the software to run Solaris services as a non-root user. (See Appendix C, “Running Services as Non-Root on Solaris.”)

- **On Windows 2000/2003 systems**: You must have Administrator privileges to install Identity Synchronization for Windows.
- 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.)

Use the installation wizard to install the Identity Synchronization for Windows Core components, as follows:

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 (Figure 3-1) so you can specify the configuration directory location.

Figure 3-1 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.

  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.

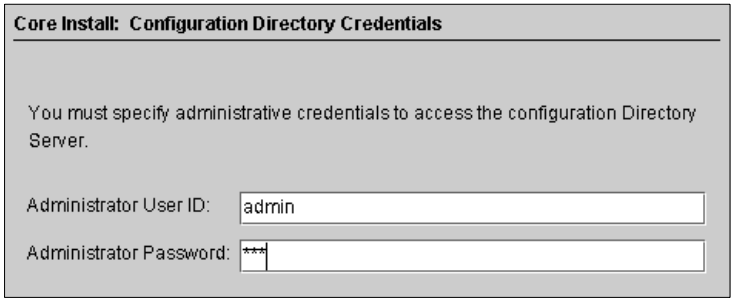
However, if you want additional transport encryption between the Console and configuration directory, be sure to enable SSL for both Administration Server and the configuration Directory Server. Then, configure a secure connection between the Administration Server to which you will be authenticating the Directory Server Console. (For information, see the *Sun Java System Administration Server 5 2004Q2 Administration Guide*).

- **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 3-2 Specifying Administrator's Credentials

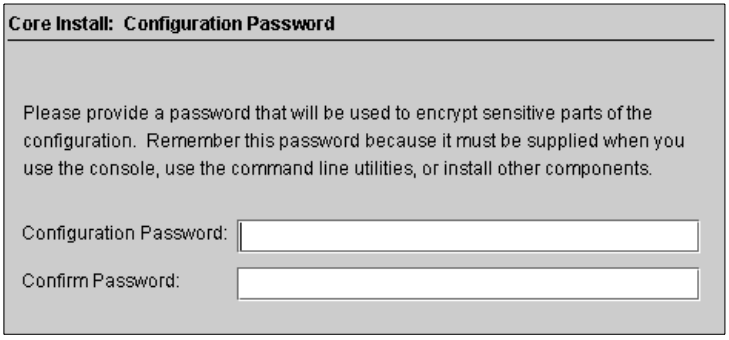


5. Enter the configuration directory Administrator's user ID and password.
   - If you specify `admin` as the user ID, you will not be required to specify the User ID as a DN.
   - If you use any other user ID, then you must specify the ID as a full DN. For example, *cn=Directory Manager*.

NOTE If you are not using SSL to communicate with the configuration directory (see Step 3, page 90), these credentials will be sent without encryption.

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

Figure 3-3 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.

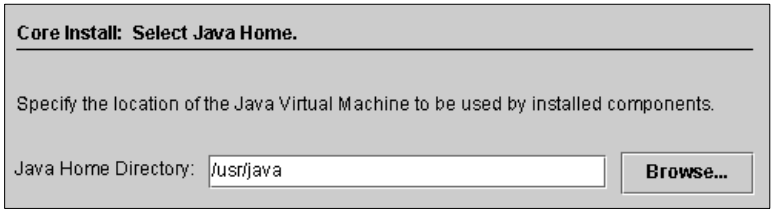
NOTE Be sure you remember this password as it will be required any time you want to

- 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 318.

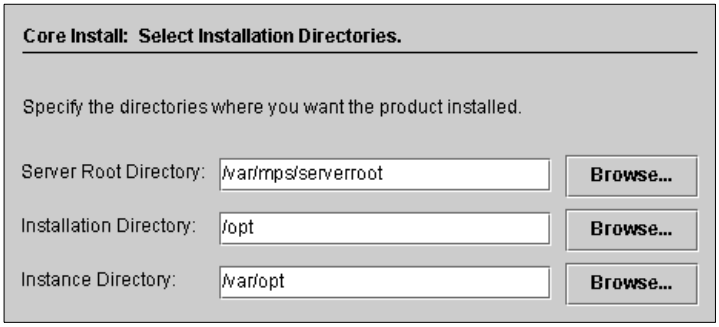
The Select Java Home panel is displayed (see Figure 3-4). The program automatically inserts the location of the Java Virtual Machine directory to be used by the installed components.

Figure 3-4 Specifying the Java Home Directory



8. Verify the Java Home Directory (must be a JDK / JRE 1.4.2_04 or later):
   - If the location is satisfactory, click Next to proceed to the Select Installation Directories panel (Figure 3-5 on page 94).
   - 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 Windows**: `C:\Program Files\j2sdk1.4.2_04`

Figure 3-5 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 Directory Server installation server root. The Console will be installed in this location.

     NOTE There is only one server root directory available on Windows operating systems, and all products will be installed in that location.

   - **Installation Directory** (*available only when you are installing Core on Solaris*): Specify the path and directory name of the installation directory. Core binaries, libraries, and executables will be installed in this directory.

   - **Instance Directory** (*available only when you are installing Core on Solaris*): Specify the path and directory name of the instance directory. Configuration information that changes (such as log files) will be stored in this directory.

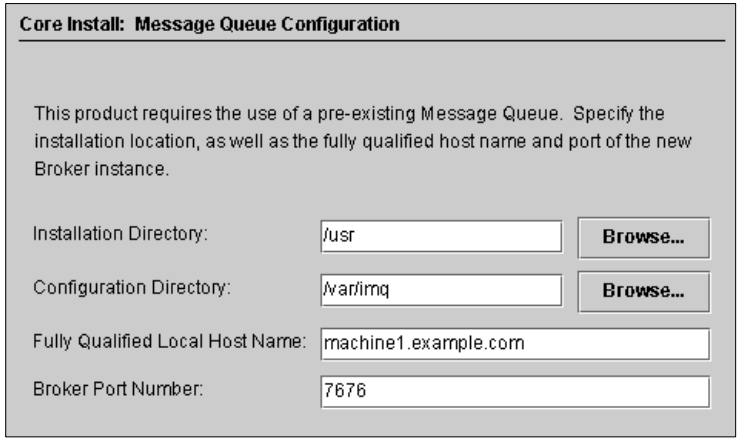
10. Click Next to proceed to the Message Queue Configuration panel.

NOTE You should have installed Message Queue 3.5 SP1 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 Windows systems: You must close any open Service Control Panel windows before continuing, or the Core installation will fail.

Figure 3-6 Configuring Message Queue



11. Enter the following information in the text fields provided or click Browse to search for and 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.)

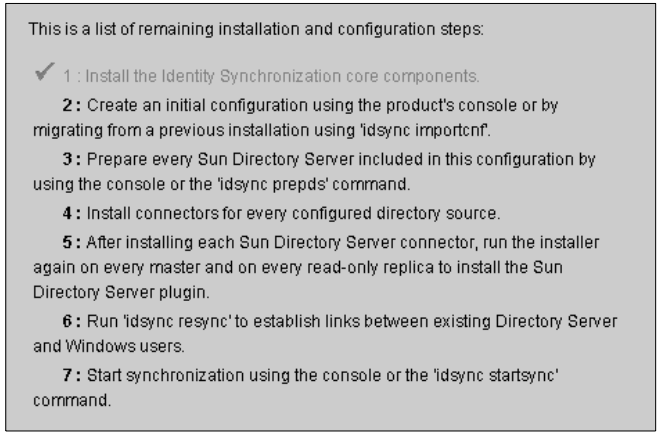
13. When the component configuration operation is complete, the Installation Summary panel is displayed to confirm that Identity Synchronization for Windows installed successfully.

    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 (Figure 3-7) 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.)

Figure 3-7 Identity Synchronization for Windows To Do List

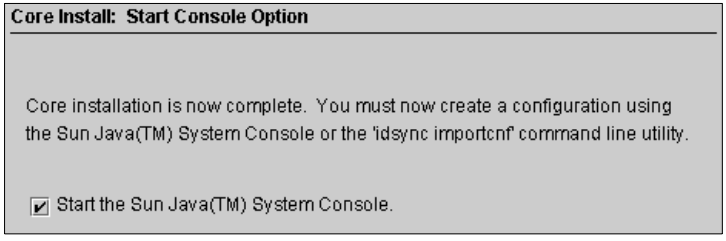


The “To Do” panel will re-display throughout the installation and configuration process. The program greys-out all completed steps in the list.

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.

Figure 3-8 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 Identity Synchronization for Windows 1 2004Q3, you can import an exported version 1.0 or SP1 configuration XML document using the `idsync importcnf` command line utility. For instructions, see Chapter 7, “Migrating to Identity Synchronization for Windows 1 2004Q3”.)

17. Click Finished.

18. If you elected to use the Console, the Sun Java System Console Login dialog box is displayed (see Figure 3-9).

Figure 3-9 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 4, “Configuring Core Resources” for instructions.

Chapter 4

Configuring Core Resources

You must initially configure the Core resources immediately after installing the Identity Synchronization for Windows Core (as described in Chapter 3).

This chapter explains how to add and configure these resources using the Console, and is organized into the following sections:

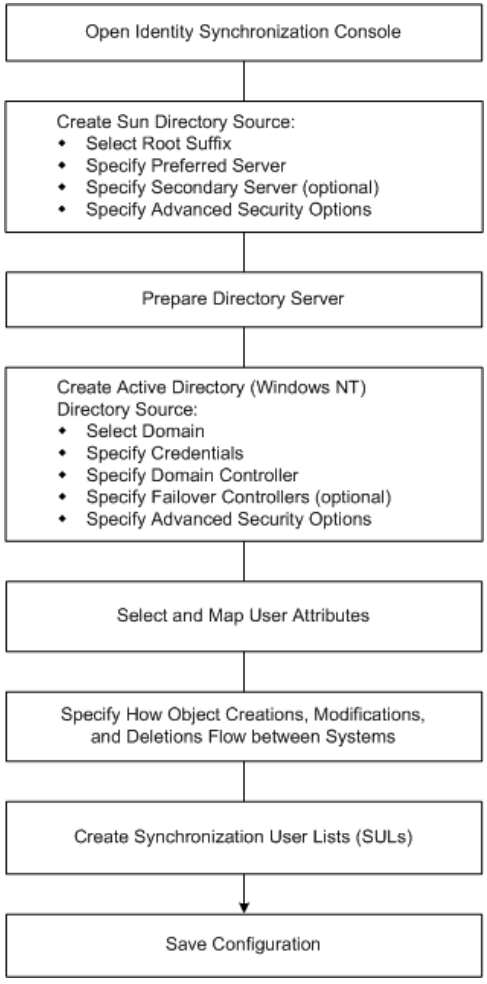
NOTE To effectively configure Core resources you must know how to configure and operate Directory Server and Active Directory.

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

Figure 4-1 illustrates the steps you will use to configure the Core resources for your deployment.

Figure 4-1 Configuring Core Resources for Your Deployment

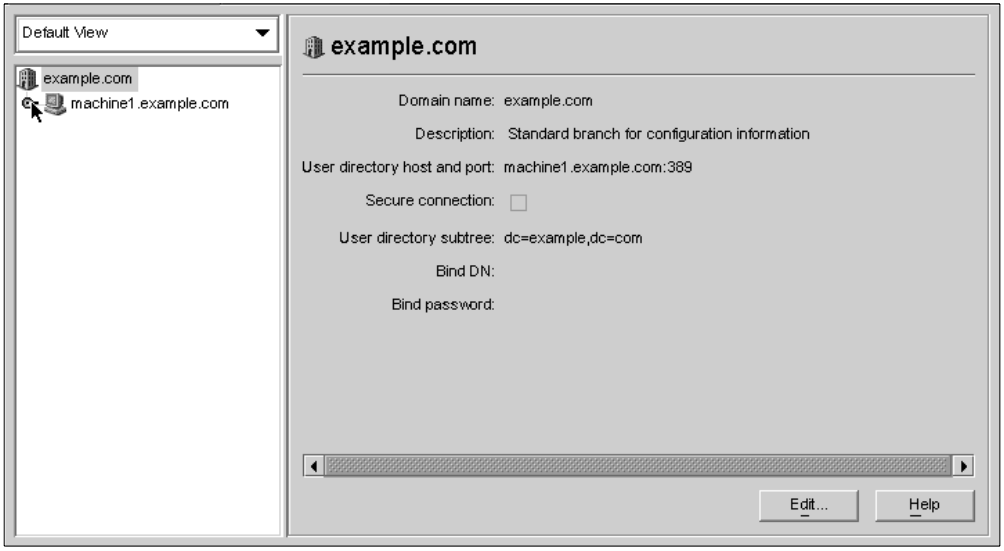


Opening the Identity Synchronization for Windows Console

NOTE If you have not logged into the Sun Java System Server Console yet, return to page 98 for instructions.

The Sun Java System Server Console window (Figure 4-2) lists all of the servers and resources under your control and provides information about your system.

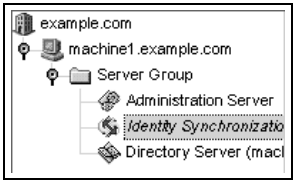
Figure 4-2 Sun Java System Server Console



To open the Identity Synchronization for Windows Console:

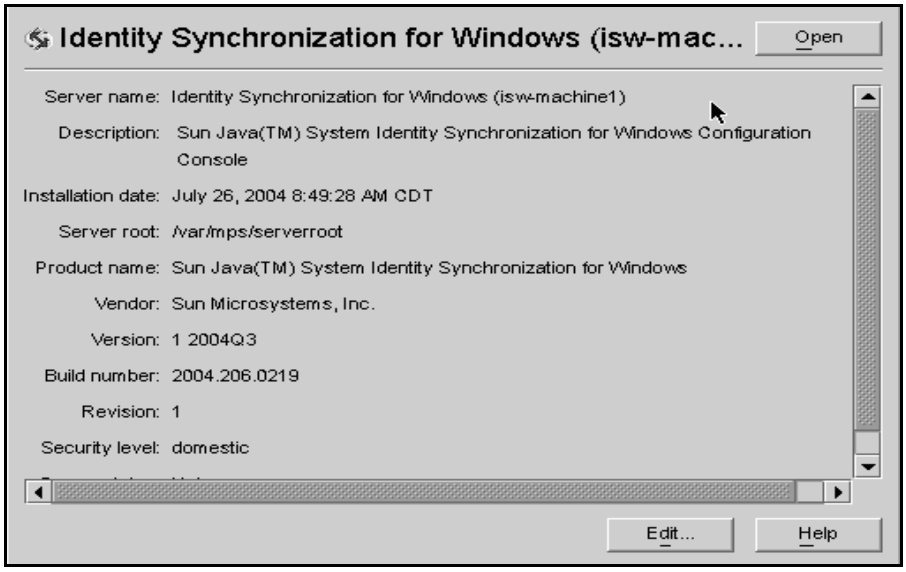
1. On the Servers and Applications tab, select the hostname node in the navigation tree that contains the Server Group to which the Identity Synchronization for Windows instance belongs.
2. Expand the Server Group node and select the Identity Synchronization for Windows node (see Figure 4-3).

Figure 4-3 Expanding the Server Group



The information panel changes to provide information about Identity Synchronization for Windows and your system (for example, see Figure 4-4).

Figure 4-4 Identity Synchronization for Windows Information Panel



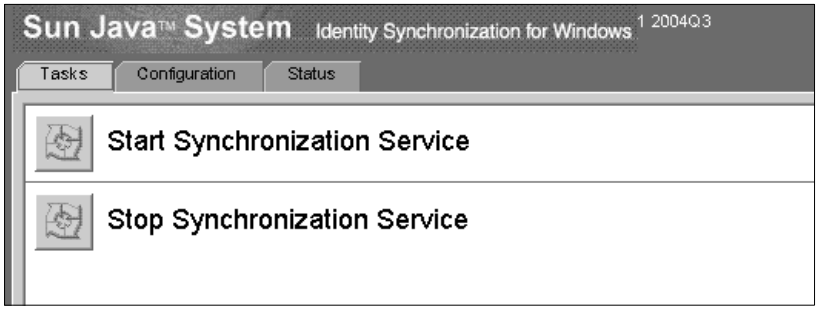
3. Click the Open button (located in the upper-right corner of the panel).

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 (see page 92). Enter the password and click OK.

   The Identity Synchronization for Windows Console is displayed, as follows:

Figure 4-5 Identity Synchronization for Windows Console: Tasks Tab



This window contains three tabs and a Status Bar:

- **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 Chapter 6, “Synchronizing Existing Users.”)

  NOTE Do not confuse starting and stopping Synchronization Services with starting and stopping Windows services.

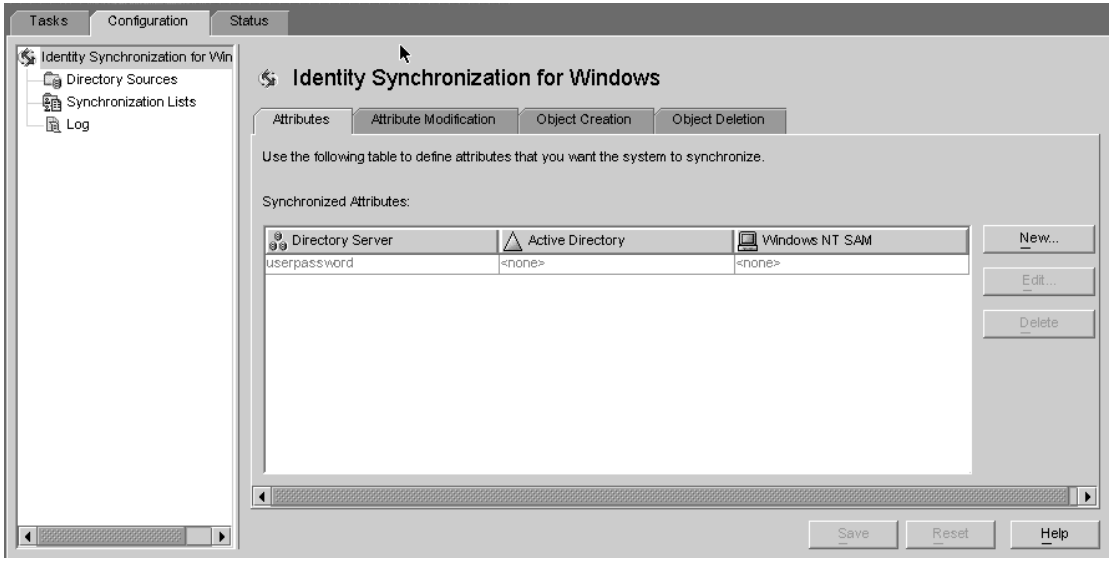
  To start or stop Windows services, you must do so from the Windows Console by selecting Start > Console > Administrative Tools > Computer Management > Services.

- **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.
  - **Status Bar**: Check this location for a brief system status.

  NOTE For more information about the Status tab, see Chapter 10.

5. Select the Configuration tab (see Figure 4-6).

Figure 4-6 Identity Synchronization for Windows 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

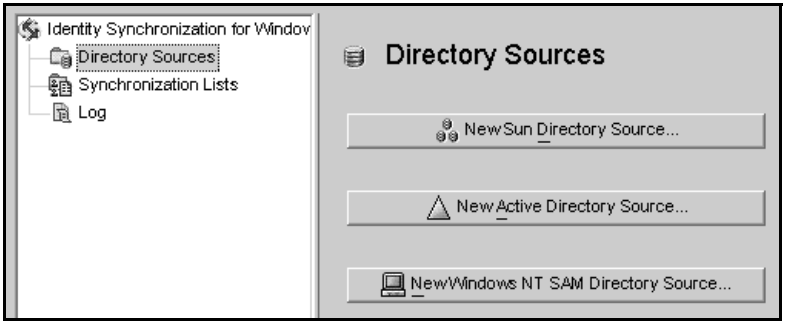
You must create directory sources in the following order (based on which sources you will be synchronizing):

1. “Creating a Sun Java System Directory Source” on page 108
2. “Preparing the Directory Server” on page 115
3. “Creating an Active Directory Source” on page 119
4. “Creating a Windows NT SAM Directory Source” on page 127

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 (see Figure 4-7).

Figure 4-7 Accessing the Directory Sources Panel



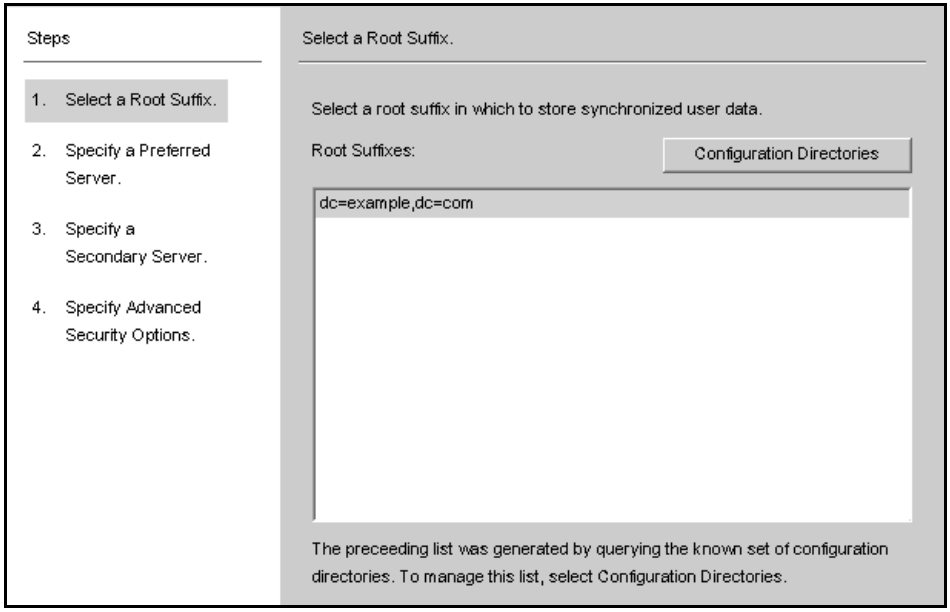
Creating a Sun Java System Directory Source

NOTE Each Sun Java System directory source is associated with a Connector and set of Plugins that can be deployed in a replication scenario of up to four masters. Any Directory Server Plugin can handle password validity checks from Windows directory sources and users can change passwords at any master; however, the Directory Server Connector is capable of synchronizing changes from Windows directory sources for up to two masters — preferred or secondary — only. Directory Server replication will replicate changes made from either of these two masters to other servers in the topology.

Use the following steps 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 4-8 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.

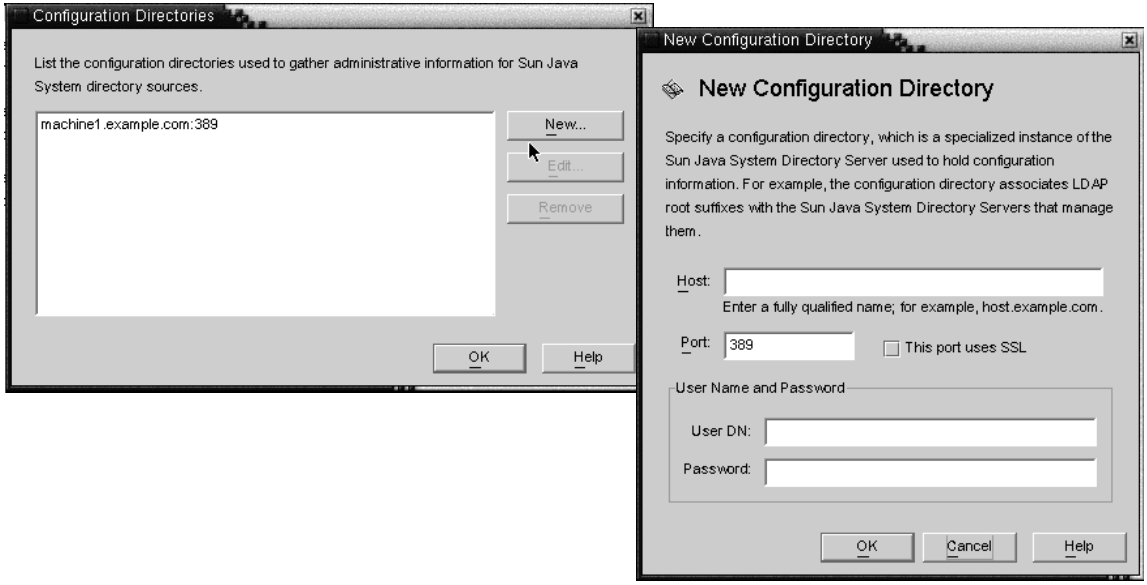
2. Select the root suffix where your users are located from the list pane. (If several root suffixes are listed, select the one where your users are located.) Click Next and proceed to Step 3.

   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 (Figure 4-9), click the New button to open the New Configuration Directories dialog box.

Figure 4-9 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.

NOTE **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:

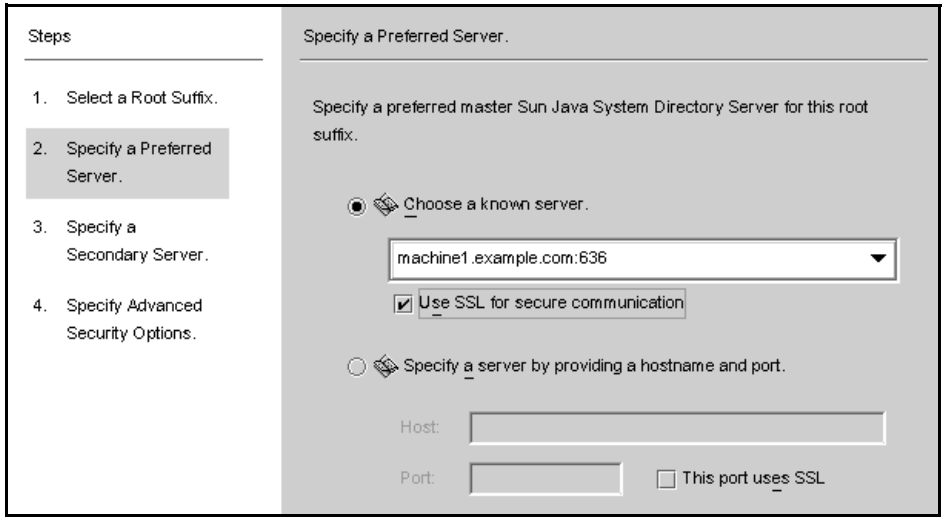
dc=<*your_machine's_DNS_domain_name*>

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 a Preferred Server panel is displayed (see Figure 4-10).

Figure 4-10 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 system.

If the preferred server fails, the secondary server can store these changes until the preferred server comes back online.

3. Use one of the following methods to select a preferred server:

   - Enable the Choose a known server button, and then select a server name from the drop-down list.

     NOTE A Directory Server must be running to appear in the list. If the server is down temporarily, enable the Specify a server by providing a hostname and port button, and then enter the server information manually.

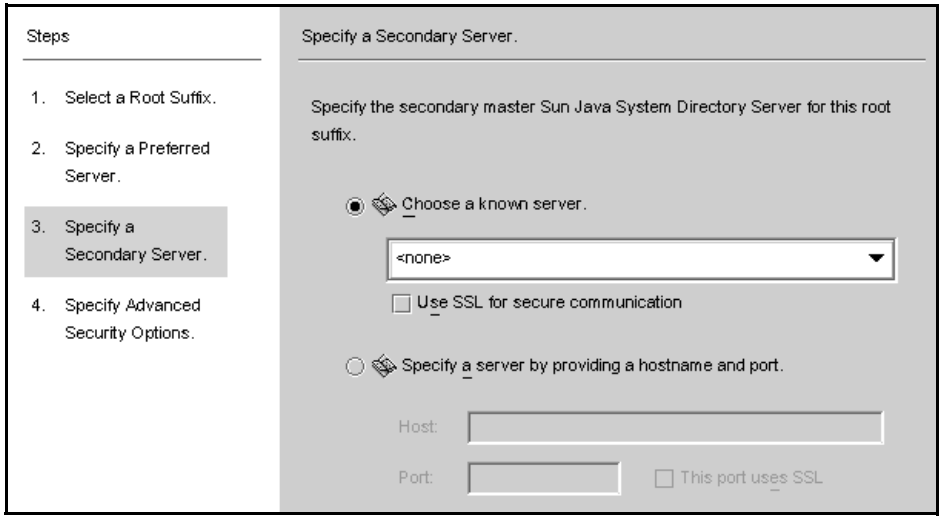
     Enable the Use SSL for secure communication box if you want the Directory Server to communicate using SSL. However, if you enable this feature there are some additional setup steps you must perform after installation. For more information, see “Enabling SSL in Directory Server” on page 299.

   - Enable the Specify a server by providing a fully qualified hostname and port button, and then type the server’s Host name and Port into the text fields provided.

     Enable the This port uses SSL box if the port you specified uses SSL.

4. Click Next and the Specify a Secondary Server panel is displayed.

Figure 4-11 Specifying a Secondary Server



- To specify a secondary Directory Server, select a name from the drop-down list or enter the information manually (use the same procedures you used to specify the preferred server), and then click Next.

NOTE The Directory Server must be running or the server name will *not* appear in the drop-down list. If the server is down temporarily, enter the server information manually.

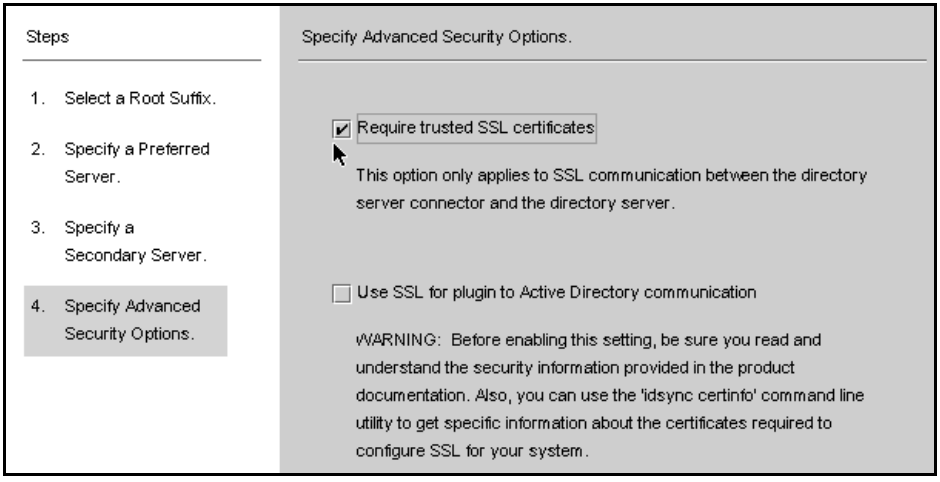
- If you do not want to use a secondary server, just click Next.

NOTE

- 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 299.

The Specify Advanced Security Options panel is displayed, as follows:

Figure 4-12 Specifying Advanced Security Options



As part of the installation process, you must install the Directory Server Plugin on each Directory Server (any master, replica, or hub) where users will bind or where passwords will be changed.

When the Directory Server Plugin synchronizes passwords and attributes to Active Directory, it must bind to Active Directory to search for users and their passwords. In addition, the Plugin writes log messages to the central log and into the Directory Server's log. By default these communications are not accomplished over SSL.

5. If you want to use secure SSL communication, *read the warning notes provided*, and then enable one or both of the following options:

   - 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 Plugin and Active Directory, enable the Use SSL for Plugin to Active Directory communication box.

NOTE

- If you enable these features, then additional setup is required after installation. See Chapter 11, "Configuring Security" for more information.
- You can use the use the `idsync certinfo` command line utility to determine which certificates you must add for each Directory Server Plugin and/or Connector certificate database. See "Using certinfo" on page 317
- If your primary and secondary Directory Servers are part of a multimaster replication (MMR) deployment, refer to Appendix E, "Installation Notes for Replicated Environments" for more instructions.

6. When you are finished with the Specify Advanced Security Options panel, click Finish.

   The program adds the selected directory sources (s) 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 5).

   - If you want to prepare the Directory Server now, click Yes to open the wizard, and then proceed to the next section, “Preparing the Directory Server” on page 115.
   - If you prefer to perform this process later, click No and proceed to “Creating an Active Directory Source” on page 119.

Preparing the Directory Server

This section explains how to prepare the Sun Java System Directory Server source for use by Identity Synchronization for Windows.

Preparing the Directory Server

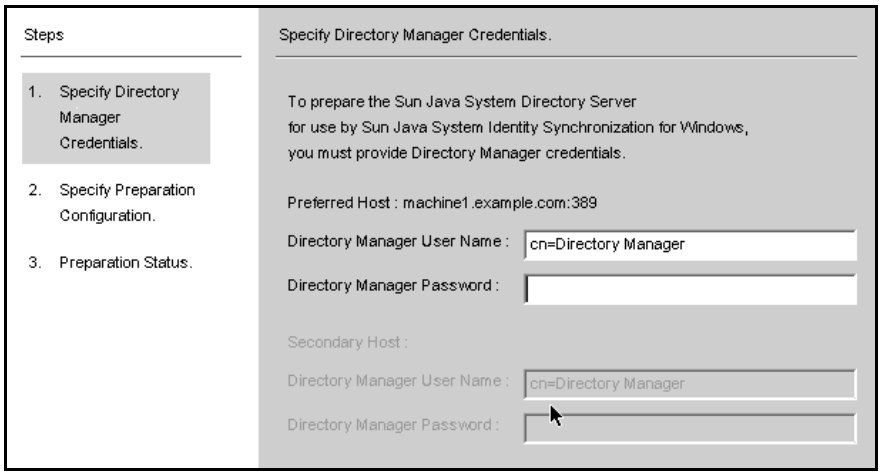
- Creates the Retro-Changelog database and access control instance available on the preferred host
- Creates the Connector user and user access control instance available on the preferred host
- Creates an equality index on the preferred and secondary hosts

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 321.
- 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 (Figure 4-13) to prepare the Directory Server.

Figure 4-13 Entering Your Directory Manager Credentials



To access this wizard, use one of the following methods:

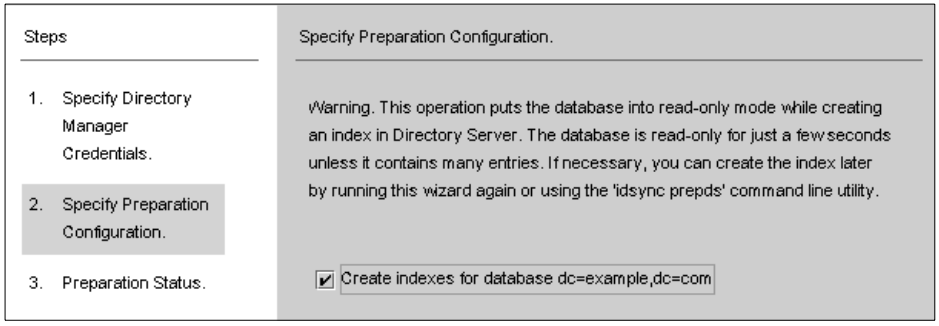
- When the Prepare Directory Server Now? dialog box is displayed, click the Yes button.
- When the Sun Directory Sources panel is displayed (on the Configuration tab), click the Prepare Directory Server button.

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 4-14).

Figure 4-14 Specifying the Preparation Configuration



Read the warning, and then decide whether to create the Directory Server indexes now or later.

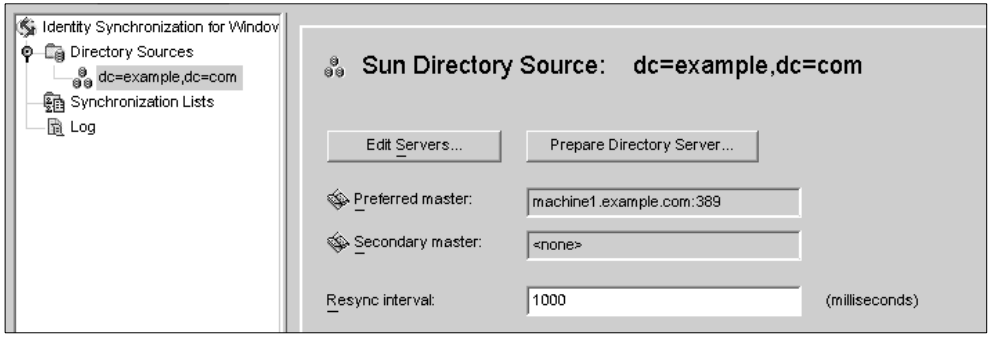
NOTE

- This operation can take a few seconds or a few minutes, 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 (see Figure 4-15).

Figure 4-15 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."
- **Prepare Directory Server**: Click this button and follow the instructions for "Preparing the Directory Server" on page 115 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.

  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.

- **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 119.
- To create a Windows NT directory source, continue to “Creating a Windows NT SAM Directory Source” on page 127.

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.

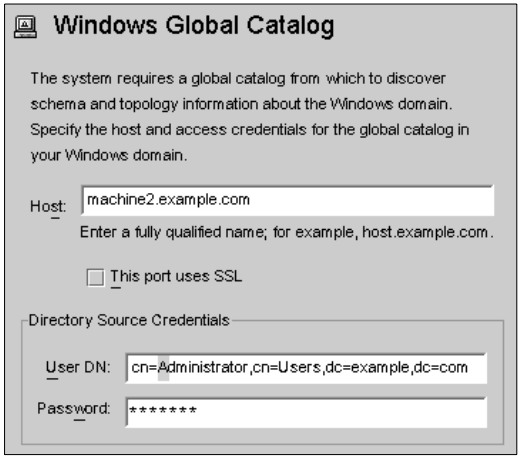
NOTE It is possible for each Active Directory server to be a global catalog and a deployment can have multiple global catalogs, but you only need to specify one global catalog.

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 4-16).

Figure 4-16 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.

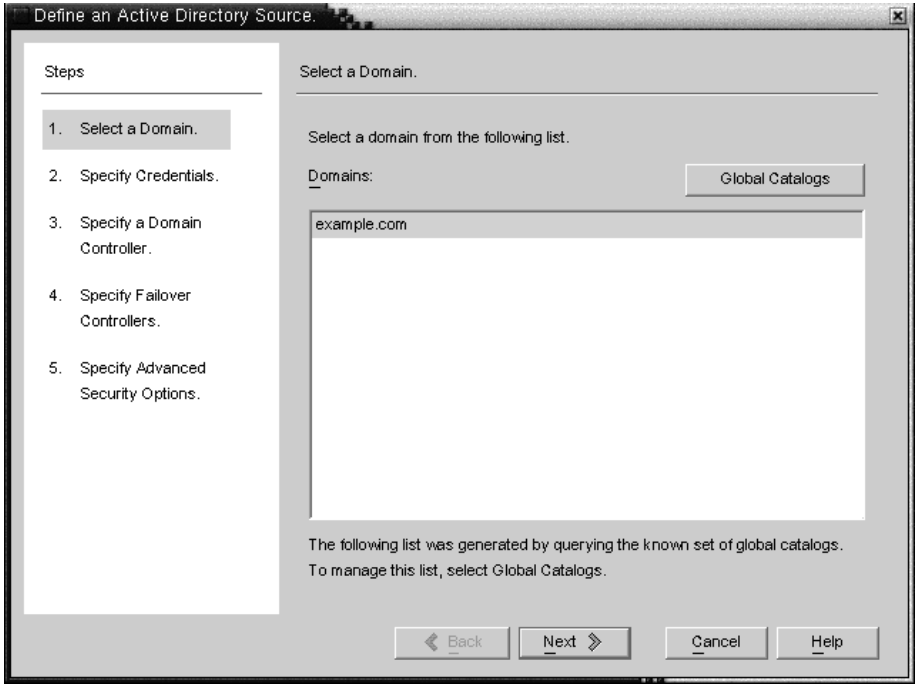
   - **User DN**: Enter your fully qualified Administrator's (bind) distinguished name. (Any credentials that enable you to browse the schemas and determine which Active Directory domains are available on your system will suffice.)

     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 4-17 Define an Active Directory Source Wizard



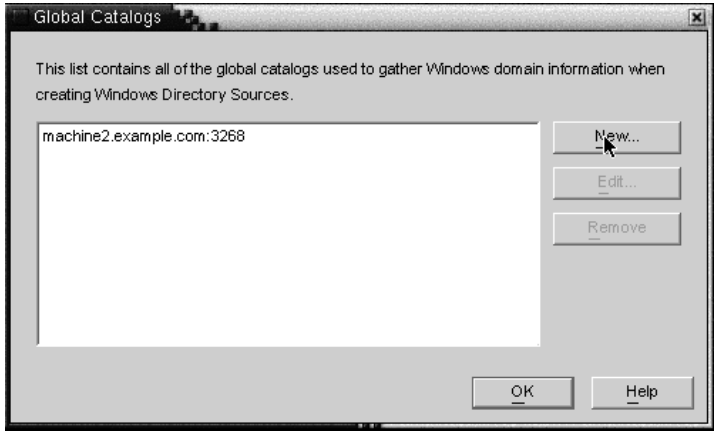
This wizard queries the Active Directory global catalog to determine what other domains exist, and displays those domains in the Domains list pane.

4. Select a name from the list pane to specify an Active Directory domain, click OK, and proceed to Step 5 on page 122.

   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 4-18).

Figure 4-18 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 on page 120), and then click OK

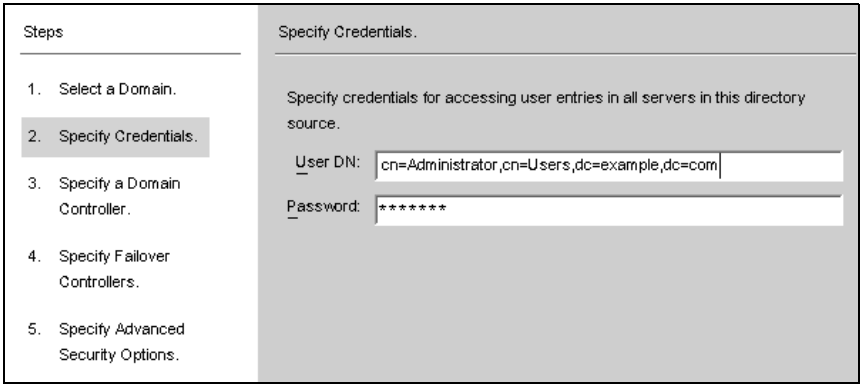
   d. The new global catalog and port, are displayed in the Global Catalogs list panel. Select the catalog name, and then click OK.

   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 4-19 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.

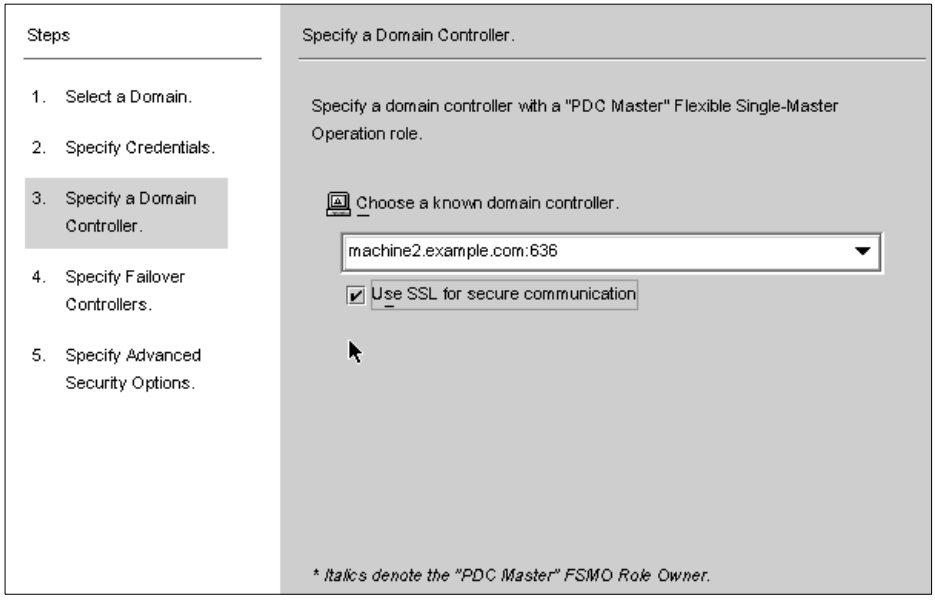
When configuring an Active Directory source, you must provide a user name and password that the Active Directory Connector can use to connect to Active Directory.

NOTE The Connector requires specific access rights. Minimum will rights 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 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 4-20 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 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.

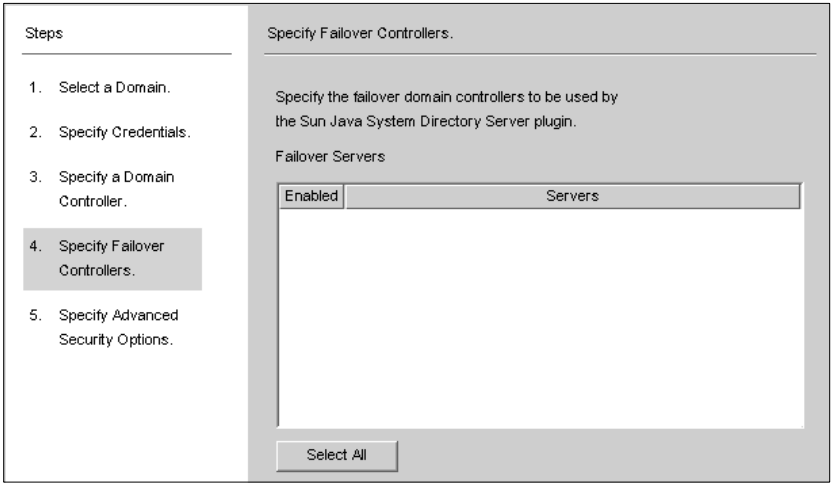
8. If you want the Identity Synchronization for Windows Connector to communicate with the domain controller over a secure port, enable the Use a Secure Port box.

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 302). Also, if you 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 Figure 4-21). You can use this panel to specify any number of failover domain controllers.

Figure 4-21 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 Plugin 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 (Figure 4-22).

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 (see Figure 4-20).

Figure 4-22 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.

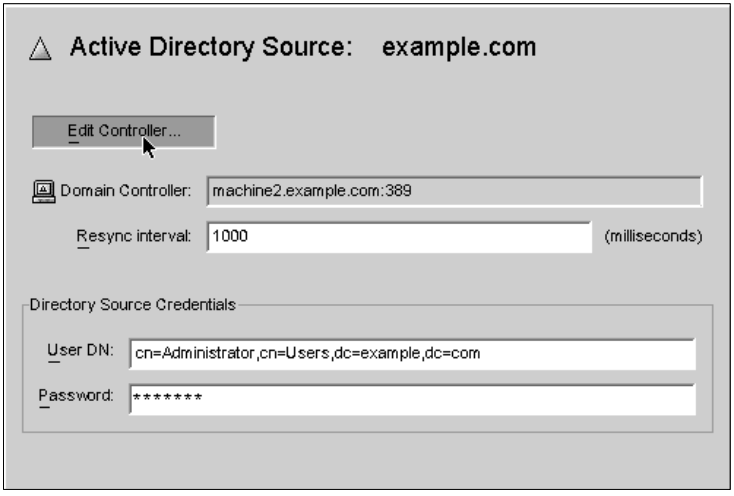
NOTE You must add Active Directory Certificates to the Active Directory Connector's certificate database. For instructions, see "Adding Active Directory Certificates to the Connector's Certificate Database" on page 305.

12. When you are finished with the Advanced Security Options panel, click the Finish button.

    The program adds the newly specified Active Directory 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 4-23).

Figure 4-23 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."
- **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.

When you are finished creating Active Directory directory sources:

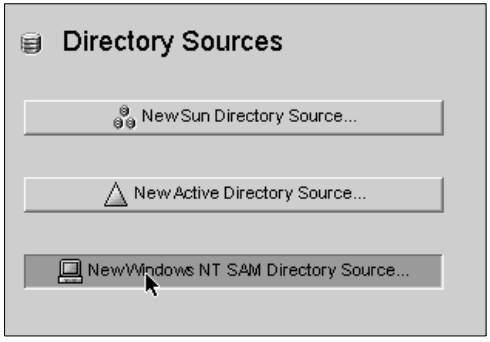
- To create a Windows NT directory source, continue to the next section, "Creating a Windows NT SAM Directory Source."
- To create and map attributes to be synchronized, continue to "Selecting and Mapping User Attributes" on page 130.

Creating a Windows NT SAM Directory Source

To deploy Identity Synchronization for Windows on a Windows NT platform, designate the NT SAM directory source as follows:

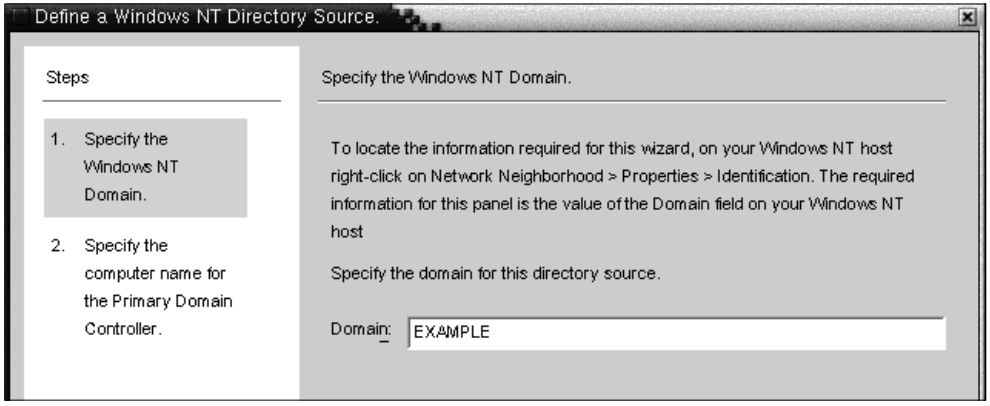
1. Select the Directory Sources node in the navigation tree, and then click the New Windows NT SAM Directory Source button.

Figure 4-24 Directory Sources Panel



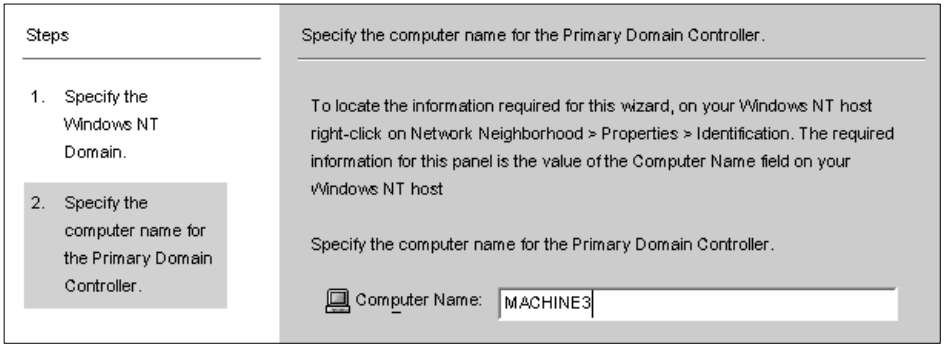
2. When the Define a Windows NT SAM Directory Source panel is displayed (see Figure 4-25), follow the instructions for locating the Windows NT domain name, and enter the unique NT directory source domain name in the Domain field. When you are done, click Next.

Figure 4-25 Specifying a Windows NT SAM Domain Name



3. When the Specify the Computer Name for the Primary Domain Controller panel is displayed (see Figure 4-26), follow the instructions for locating the Primary Domain Controller computer name, and enter the information in the Computer Name field.

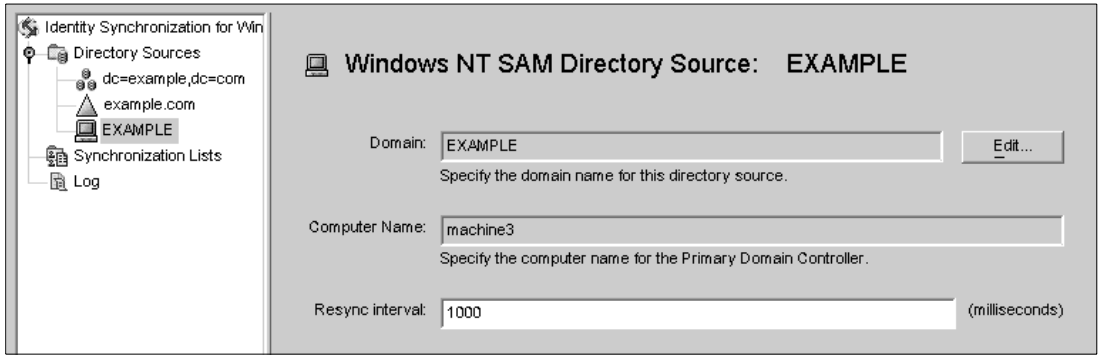
Figure 4-26 Specifying a Name for the Primary Domain Controller



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 (see Figure 4-27).

Figure 4-27 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."
- **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 130 for instructions.

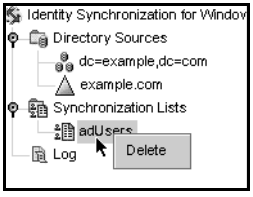
Deleting Directory Sources

NOTE If you have already installed the connector associated with a directory source, you must uninstall the connector *before* you delete the directory source.

If you must delete a directory source, use the following steps:

1. Before you can delete the directory source, you must first delete all of the Synchronization User Lists (SULs) associated with that source.
   a. Right-click the affected Synchronization User List node listed under Synchronization List in the navigation tree.
   b. When the pop-up menu displays, select Delete to remove the SUL.

Figure 4-28 Deleting a Synchronization User List



2. Right-click the directory source node listed under Directory Sources in the navigation tree.
3. When the pop-up menu displays, select Delete to remove the directory source.

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.

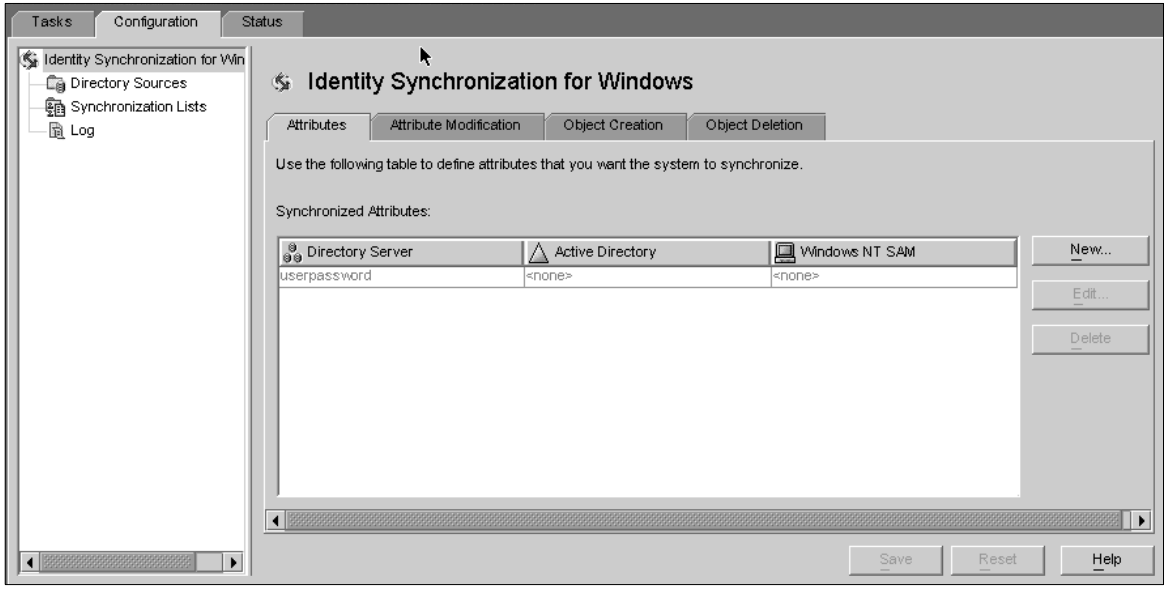
  Some creation attributes are *mandatory* based on the schema used for each platform. These attributes are required for password synchronization and they must be mapped to Sun attributes to successfully create a user object class entry on the Active Directory server.

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 (see Figure 4-29).

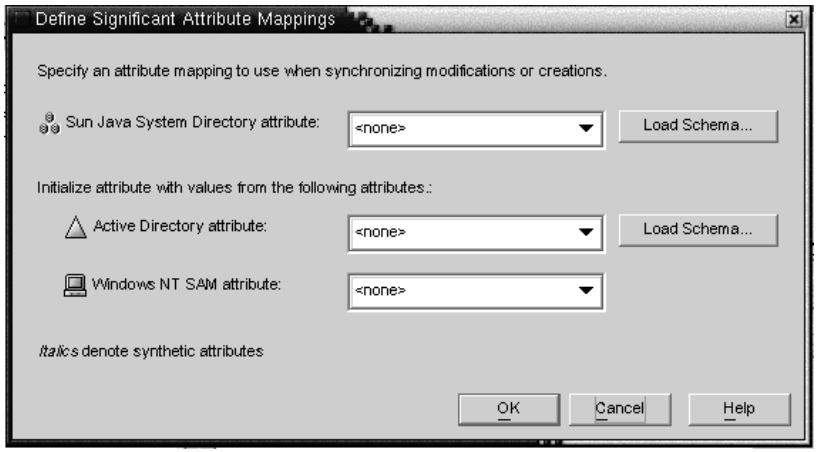
Figure 4-29 Attributes Tab



2. Select the Attributes tab and then click the New button.

   The Define Significant Attribute Mappings dialog box is displayed (Figure 4-30). Use this dialog box to map attributes from Directory Server to your Windows Systems (Active Directory and/or Windows NT).

Figure 4-30 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.

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 Step 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.

Figure 4-31 Completed Synchronized Attributes Table

Directory Server	Active Directory	Windows NT SAM
userpassword	unicodepwd	user_password
cn	cn	<none>
telephonenumber	telephonenumber	<none>

NOTE The program automatically uses *inetOrgPerson* as the default objectclass for Sun Java System 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 134 for instructions.

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=%B%`, 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%`..)

Changing the Schema Source

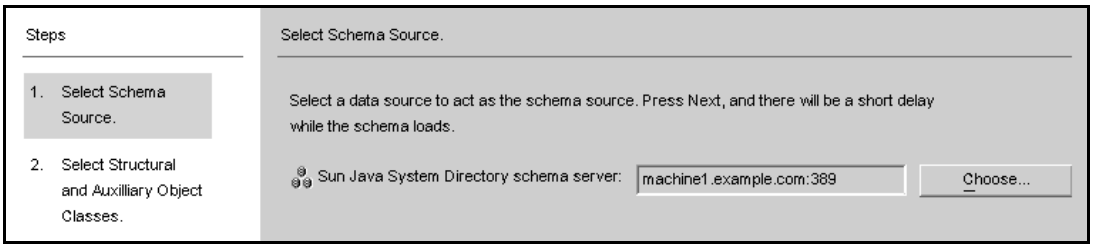
The program automatically provides default schema sources, but allows you to change the default schema.

Use the following procedure 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 4-32).

Figure 4-32 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.

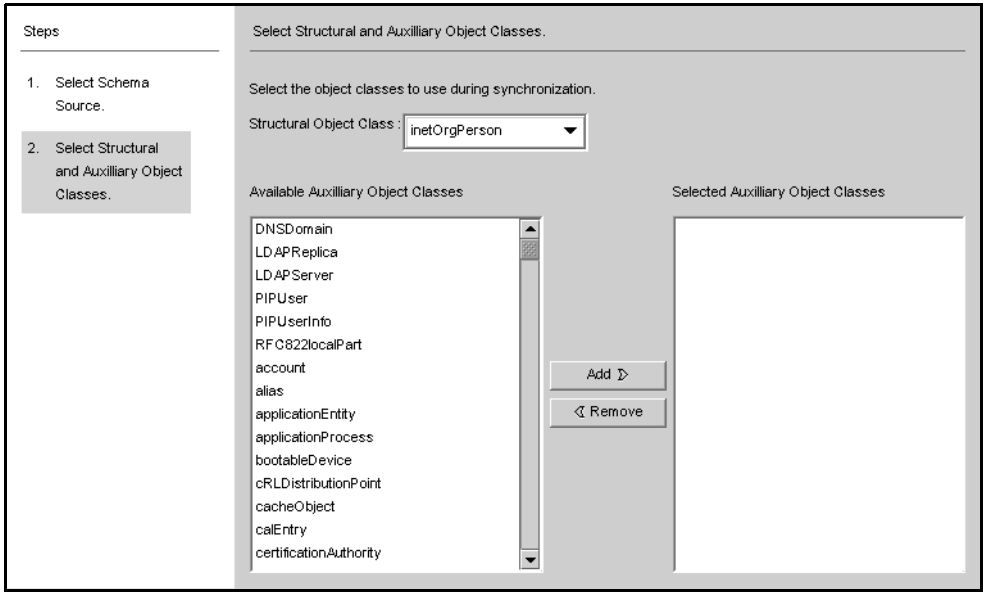
Click Edit, and when the Edit Configuration Directory dialog box displays, you can change the Host, Port, User DN, and/or Password. Click OK when you are done.

- 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 4-33).

Figure 4-33 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 Sun and Windows Systems.

By default, Identity Synchronization for Windows

- Synchronizes from Windows to Sun Java System 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:

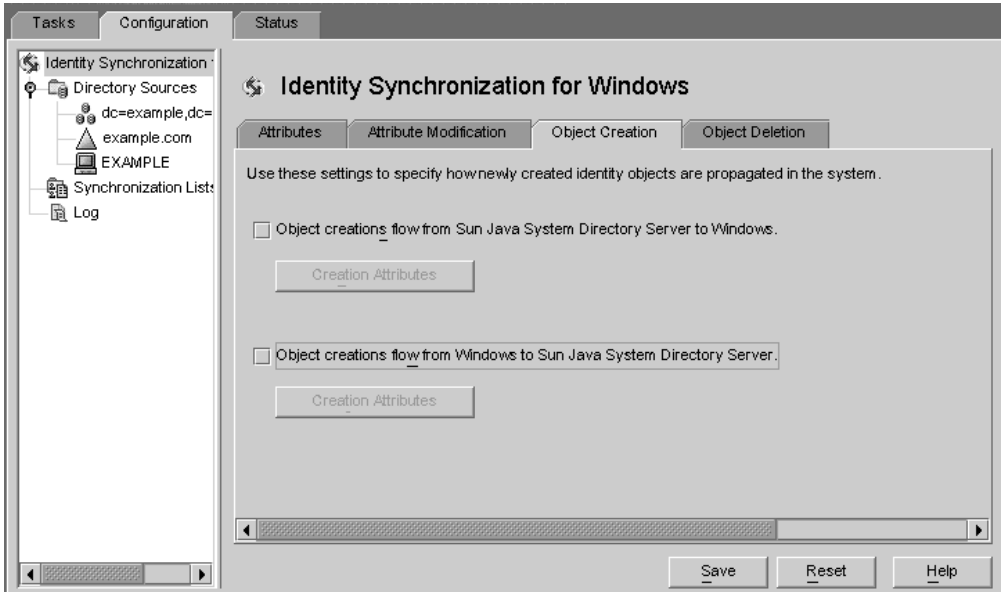
- “Specifying How Object Creations Flow” on page 137
- “Specifying How Object Modifications Flow” on page 142
- “Specifying How Deletions Flow” on page 152

Specifying How Object Creations Flow

Use the following steps to specify how object creations should flow between Directory Server and Active Directory systems:

1. Click the Object Creation tab.

Figure 4-34 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 (see Figure 4-35 and Figure 4-36).

Figure 4-35 Creation Attributes Mappings and Values: Directory Server to Windows

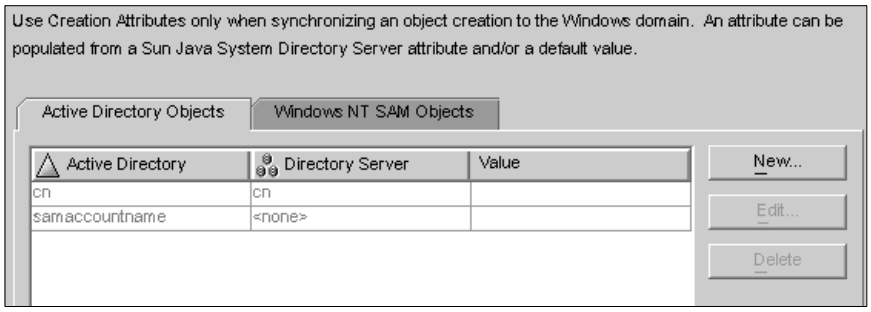
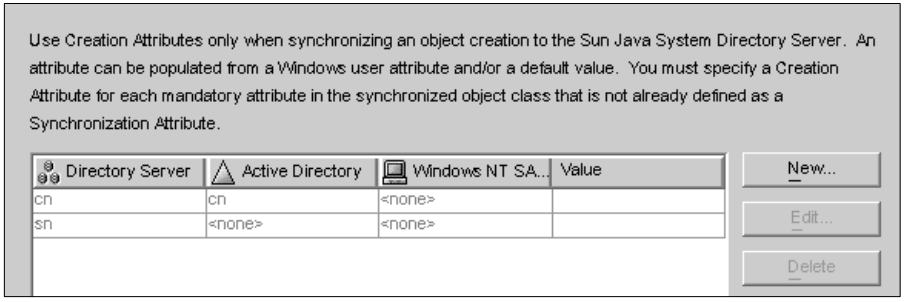


Figure 4-36 Creation Attributes Mappings and Values: Windows to Directory Server



You can use either dialog box to do the following:

- Specify new creation attributes (page 139)

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 130.

- Edit existing attributes (see page 139)
- Remove existing attributes (see page 139)

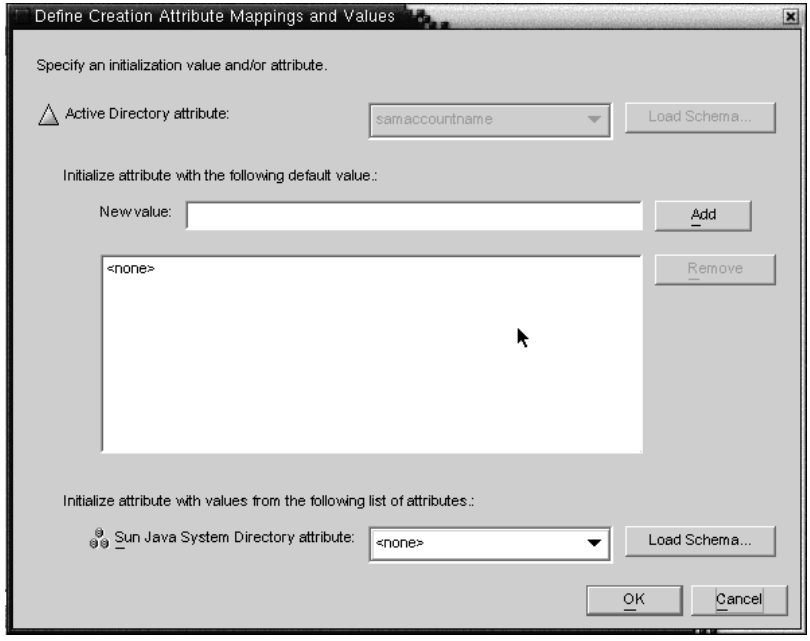
Specifying New Creation Attributes

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.)

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 4-37).

Figure 4-37 Defining Creation Attribute Mappings and Values



2. Select an attribute value from the Active Directory attribute drop-down list.

Figure 4-38 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.

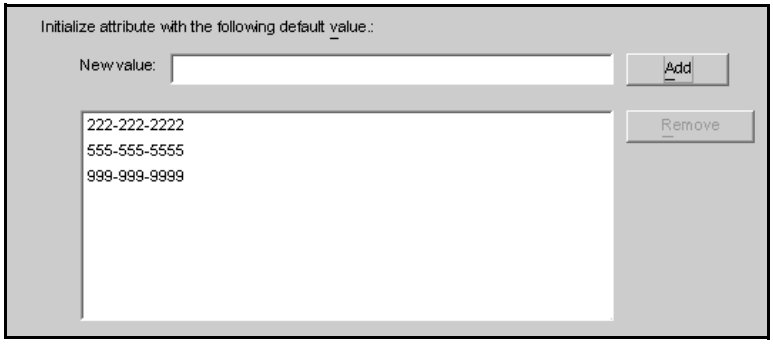
For example, if your company has three fax telephone numbers, you can specify the `facsilimiletelephonenumber` attribute for both Sun Java System Directory Server and Active Directory, and specify the three numbers.

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.

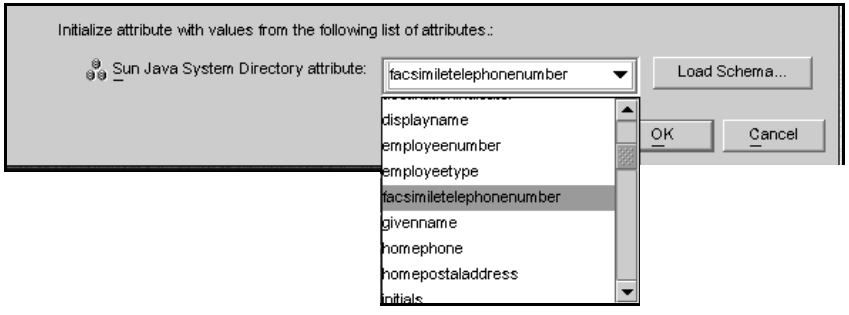
   The program adds the attribute value to the list pane. Repeat this step as many times as necessary to add multiple attribute values.

Figure 4-39 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 4-40 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 4-41 Completed Creation Attributes and Mappings Table

Active Directory	Directory Server	Value
cn	cn	
samaccountname	<none>	
facsimiletelephonenumber	facsimiletelephonenumber	[222-222-2222,555-555-55...

6. To designate additional attributes, repeat these steps.

Editing Existing Attributes

To edit any of the 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 mappings or values,

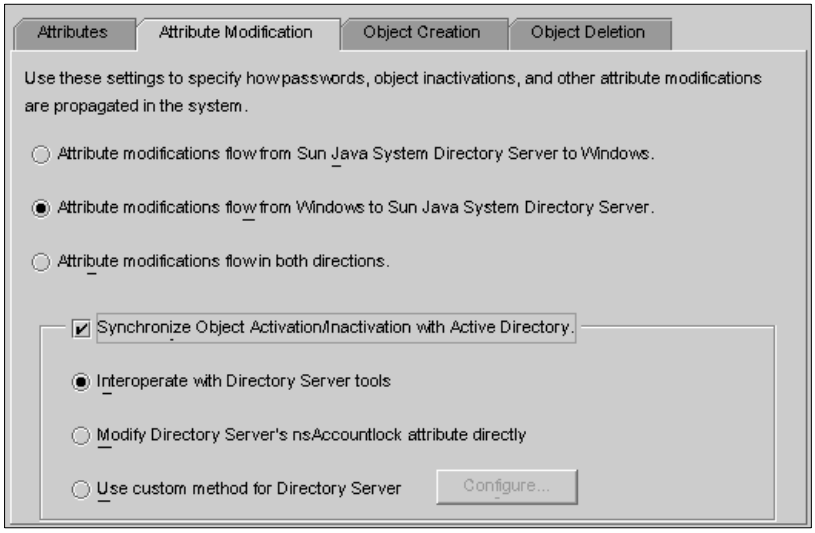
1. Select the Object Creation tab, and click the Creation Attributes button located under the selected creation option.
2. When the Creation Mappings and Values dialog box is displayed, select the attribute from the table, and then click the Delete button.

   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 (Figure 4-42) to control how modifications made to user attributes and passwords will be propagated (flow) between your Sun and Windows systems.

Figure 4-42 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.

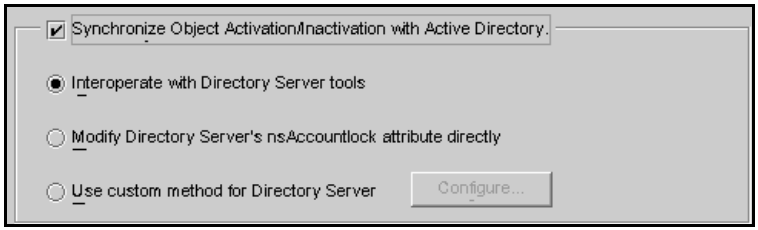
- **Attribute modifications flow from Sun Java System Directory Server to Windows**: Propagates changes made in the Directory Server environment to your Windows servers.
- **Attribute modifications flow from Windows to Sun Java System Directory Server** (*Default*): Propagates changes made in the Windows 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 (see Figure 4-43) 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.

Figure 4-43 Synchronizing Object Activations and Inactivations



To synchronize object activations / 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:
   - **Interoperate with Directory Server tools** (see page 145)
   - **Modify Directory Server's nsAccountLock attribute directly** (see page 146)
   - **Use custom method for Directory Server** (see page 146)

NOTE These options are mutually exclusive.

- Enable the Interoperate with Directory Server Tools option, and 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*`>`.
- Enable the Modify Directory Server's `nsAccountLock` attribute option and Identity Synchronization for Windows will not detect objects that are activated / inactivated using the Directory Server Console or command line utilities.
- Enable the Use custom method for Directory Server option and Identity Synchronization for Windows cannot lock objects out of the directory unless access to the directory is controlled by an external application, such as Sun Java™ System Access Manager (formerly Sun JES Identity Server).

Interoperating with Directory Server Tools

Select this option if you use the Directory Server Console or command line tools to activate/inactivate an object.

- 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>`.

Table 4-1 describes how Identity Synchronization for Windows detects and synchronizes object activations/inactivations when you enable the Interoperate with Directory Server Tools option:

Table 4-1 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`. 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.

Table 4-2 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 4-2 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 inactivation from 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).

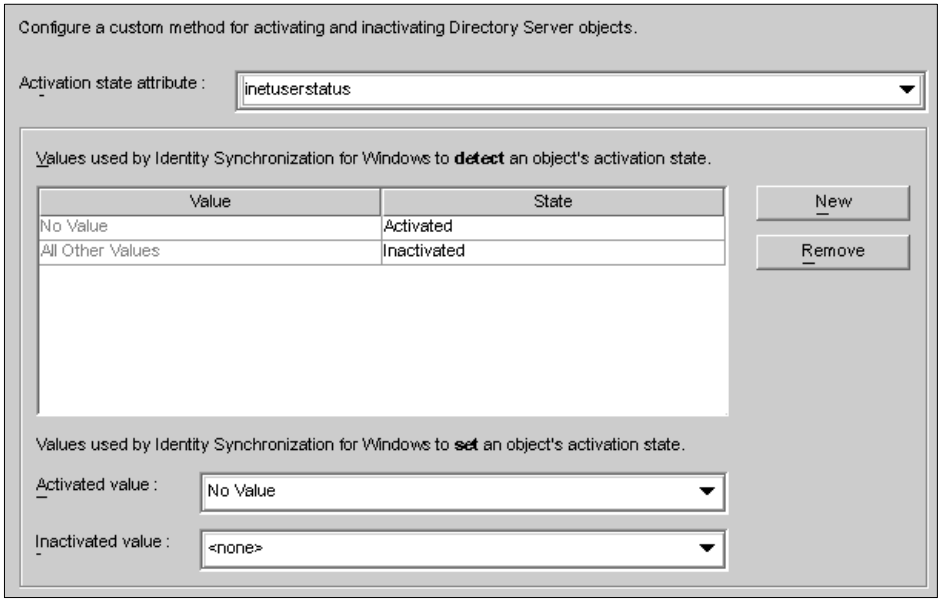
When you configure a custom method for Directory Server, you must specify

- 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 (see Figure 4-44).

Figure 4-44 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.

Table 4-3 Specifying Activated and Inactivated States

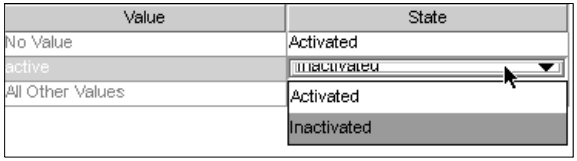
Value	State	Result
No Value	Activated	If the attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as activated.
	Inactivated	If the attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as inactivated.
***<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 Use the following procedure 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 4-45 Selecting a State



For example, if you were using Access Manager:

1. Select the **`inetuserstatus`** attribute from the Activation state attribute drop-down list.
2. Click the New button and enter **`active`**, **`inactive`**, and **`deleted`** attribute values to the Value column of the table.
3. 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, Table 4-4 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).

Table 4-4 Example Results Using inetuserstatus Values

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.

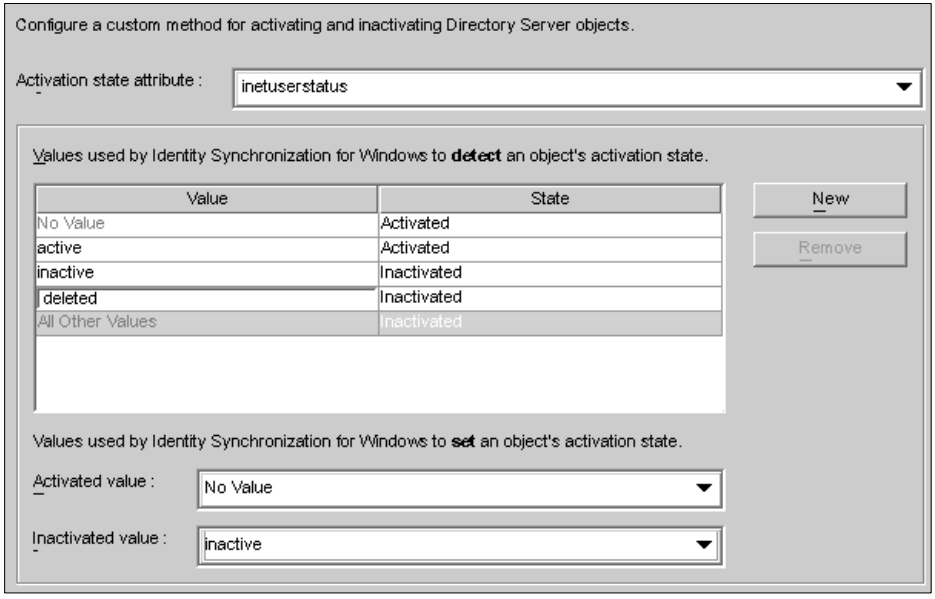
- **Activated value**: Controls the object's active state.
  - **No Value**: If the object contains the active value, Identity Synchronization for Windows will set the state to activated in Directory Server.
  - **active**: If the object contains the active value, Identity Synchronization for Windows 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.

Figure 4-46 illustrates a completed Configure Custom Method for Directory Server dialog box.

Figure 4-46 Example: Completed Dialog



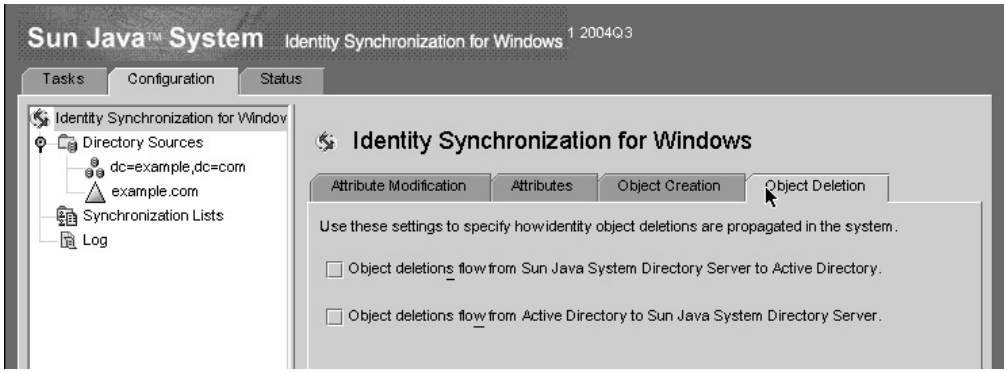
Specifying How Deletions Flow

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.

1. Select the Identity Synchronization for Windows node at the top of the navigation pane, and then click the Object Deletion tab.

Figure 4-47 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 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 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 two directory sources 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, “Defining and Configuring Synchronization User Lists” on page 341.

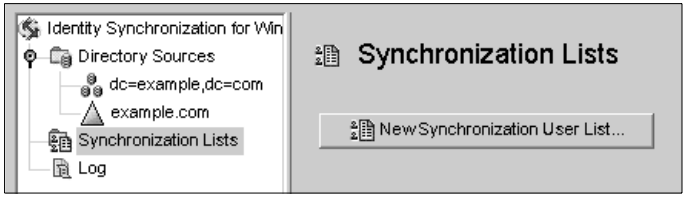
Both of the SUL elements contain three definitions that identify which users to synchronize:

- **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:

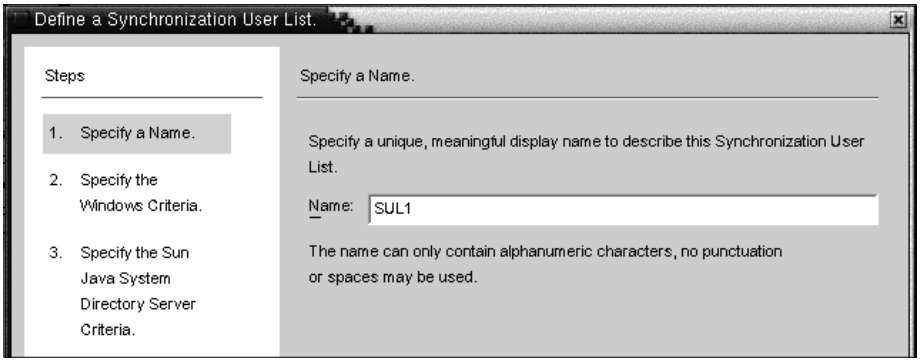
1. Select the Synchronization User Lists node in the navigation tree, and then click New Synchronization User List button.

Figure 4-48 Creating a New Synchronization User List



The Define a Synchronization User List wizard is displayed, as follows:

Figure 4-49 Specifying a Name for Your SUL



The program default for your first Synchronization User List is *SUL1*.

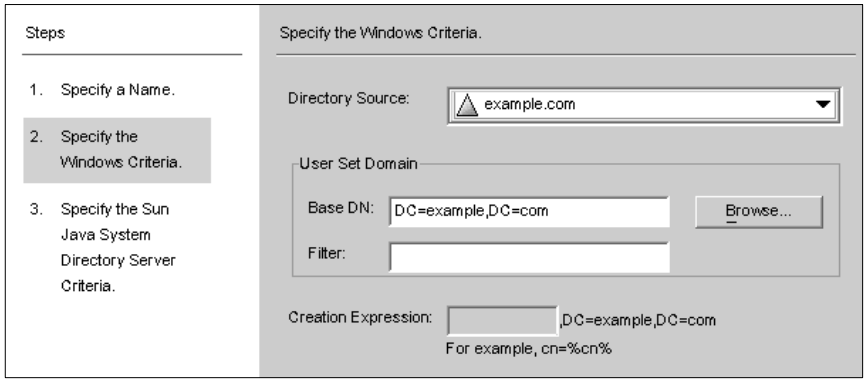
- If the default name is acceptable, click Next.
- If you want to use a different name, type a different name into the Name field and then click Next.

NOTE

- 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, as shown in Figure 4-50.

Figure 4-50 Specifying the Windows Criteria



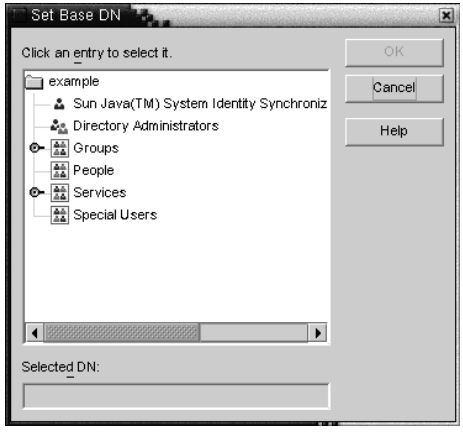
2. Select a Windows Directory Source from the drop-down list.

NOTE You cannot edit this directory source *after* creating the SUL.

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.

Figure 4-51 Selecting 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.

4. You can enter an equality, a presence, or a substring Filter to specify which users in this 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 (See “Specifying How Object Creations Flow” on page 137).

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:

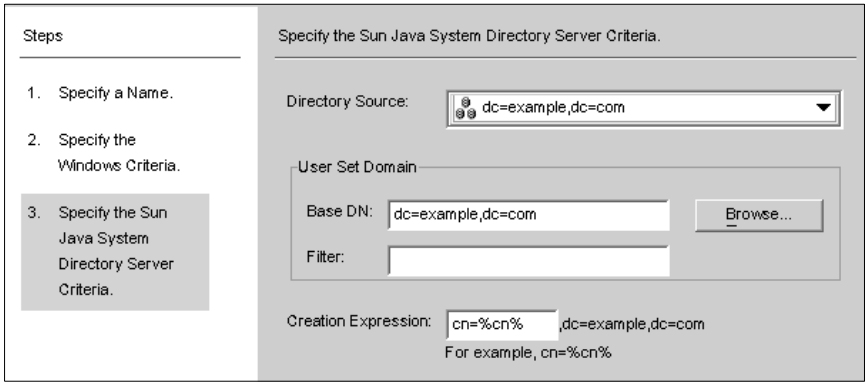
   cn=*%cn%*,cl=*users*,dc=example,dc=com

   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 4-52 Specifying Directory Server Criteria

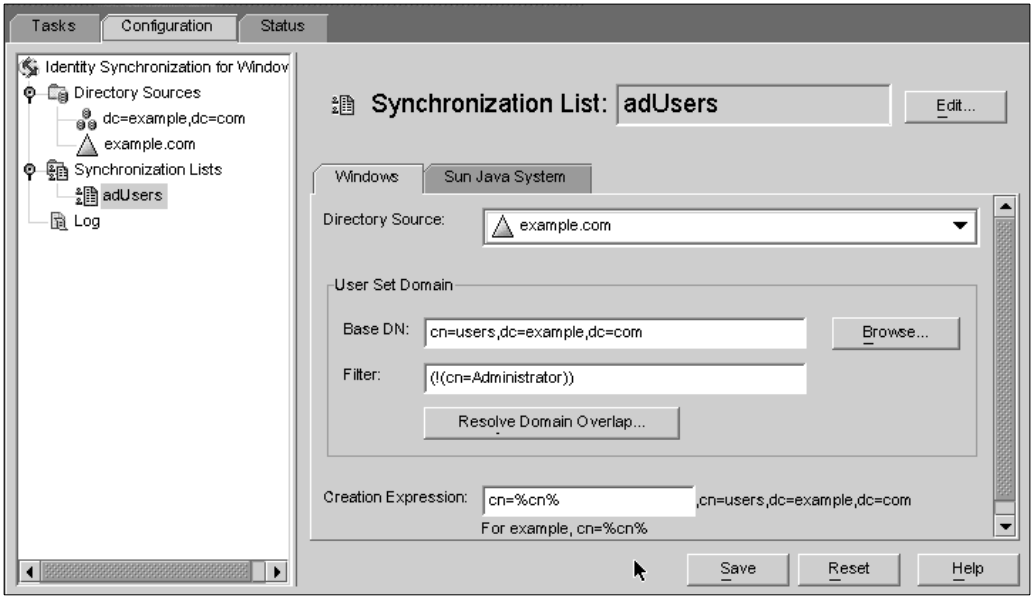


NOTE You cannot edit the Active Directory or Directory Server directory sources included in this SUL after you click the Finish button to create the SUL.

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 4-53 Synchronization List Panel



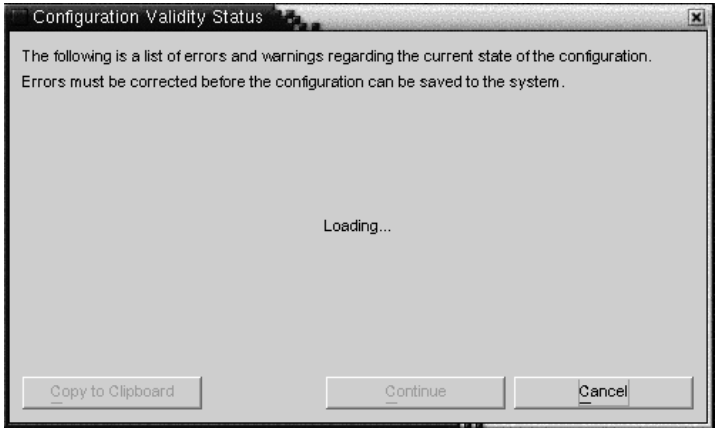
10. In cases where a user matches multiple lists, click the Resolve Domain Overlap button to define a preference for the synchronization user list. (For more information, see “Understanding Synchronization User List Definitions” on page 341.)
11. Create a Synchronization User List that includes every directory source in your network except for the Directory Server.

Saving a Configuration

To save your current configuration from any of 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 4-54 Configuration Validity Status Window



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.

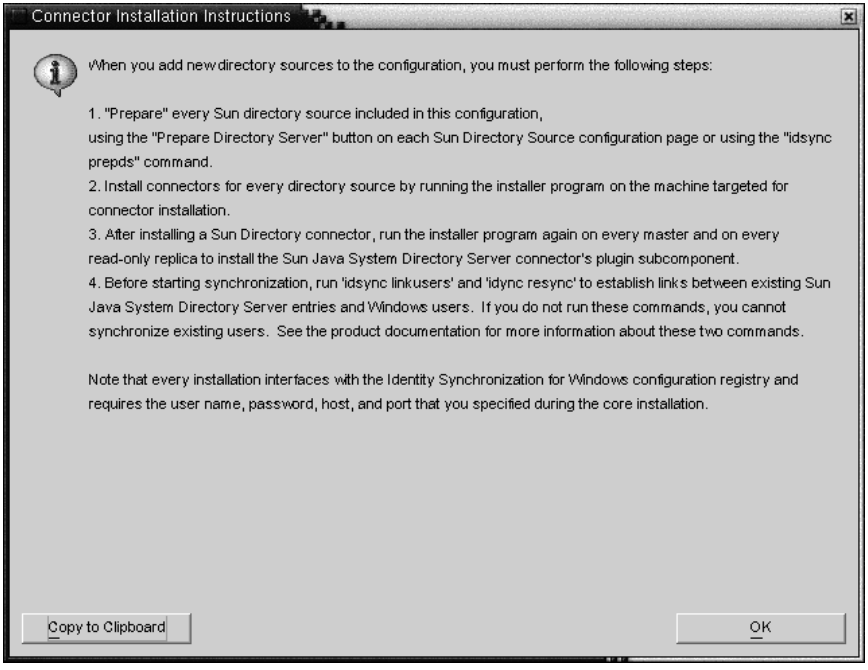
- You cannot save a configuration with errors.
- You can save configurations with warnings, but it is better to try and clear the warnings first.

3. If your configuration is valid, click Continue to save the configuration.

   A Connector Installation Instructions dialog box is displayed (similar to the list in Figure 4-55), 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 4-55 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 5, “Installing Connectors and Directory Server Plugins” for instructions.

Chapter 5

Installing Connectors and Directory Server Plugins

This chapter provides instructions for installing the Identity Synchronization for Windows Connectors and Directory Server Plugins. 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 / Directory Server Plugin installation 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 or the Plugin, the program perceives a conflict about which component is adding configuration data to the server and generates an error message.
- You must install the Directory Server Plugin on every Directory Server machine in your deployment that stores users to be synchronized; which includes masters, replicas, and hubs.
- 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
    - The installation directory
- You must run the installation program each time you install a Connector or a Directory Server Plugin.

  For example, if you are installing a Directory Server Connector, a single Directory Server Plugin, and an Active Directory Connector, you will run the installation program three separate times after Core is installed.

Running the Installation Program

Use the following procedure to restart and run the installation program. You will repeat these steps each time you install a Connector or a Directory Server Plugin:

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.

     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 Step 4.

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 page 93).

8. When you are finished, click Next.

NOTE At this point, the installation process becomes specific to the Directory Server Plugin or the type of Connector you are installing.

- To install a Connector, proceed to "Installing Connectors" on page 164.
- To install a Directory Server Plugin, proceed to "Installing Directory Server Plugins" on page 175.

Installing Connectors

This section explains how to install the three types of Identity Synchronization for Windows Connectors, as follows

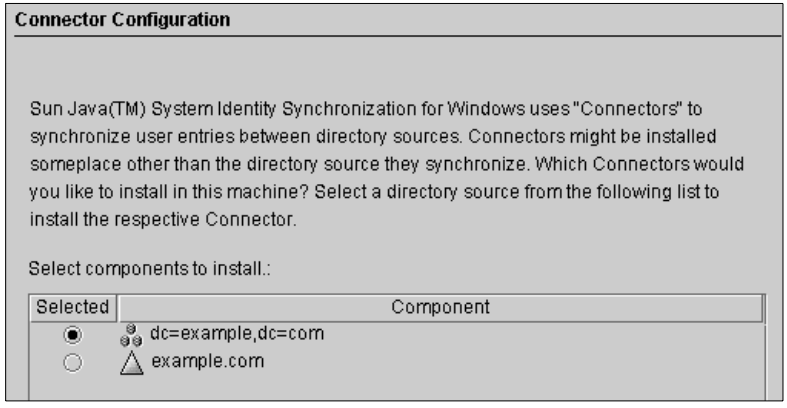
- "Installing the Directory Server Connector" on page 165
- "Installing an Active Directory Connector" on page 170
- "Installing the Windows NT Connector" on page 174

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 “Running the Installation Program” on page 162, the Connector Configuration panel displays.

Figure 5-1 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 in Figure 5-1), the program will remove the entry from the list pane.

The following table contains some example directory source entries:

Table 5-1 Directory Source Examples

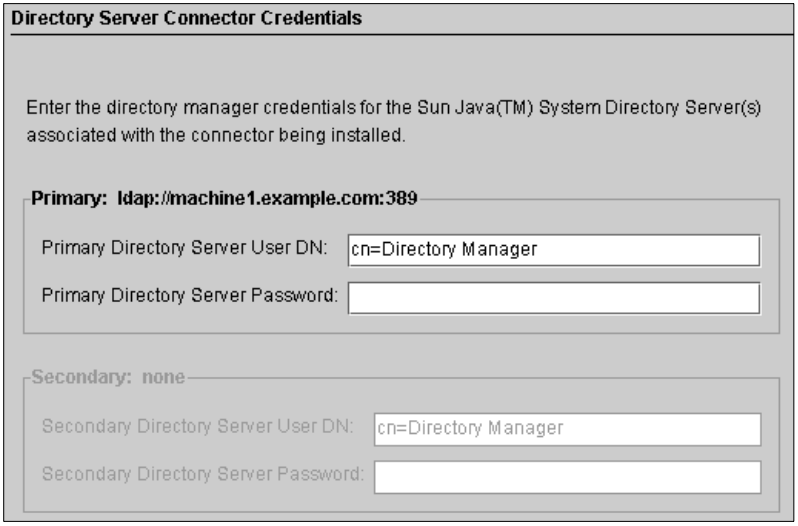
Directory Source	Example Entry
Sun Java System Directory Server	`dc=example,dc=com`
Windows Active Directory	`example.com`
Windows NT SAM	`EXAMPLE`

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 (Figure 5-2).

Figure 5-2 Entering Directory Server Connector Credentials



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:

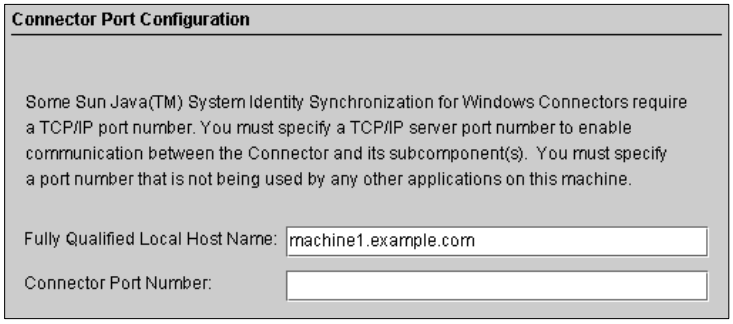
- **Primary Directory Server User DN**: If necessary, change the default user DN by entering a fully qualified Directory Manager distinguished name.
- **Primary Directory Server Password**: Enter your Directory Manager password.

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 (page 115), 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.

Figure 5-3 Specifying the Connector Local Host and Port

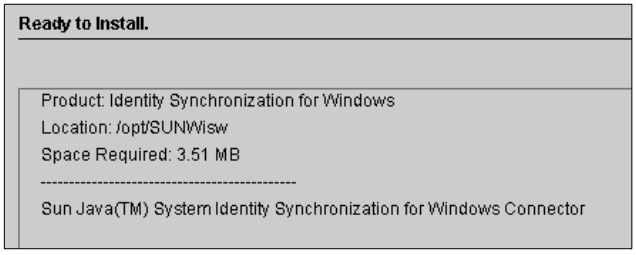


3. Enter the Fully Qualified Local Host Name with the domain and an available port number where the Connector will listen. (Specifying a port already in use will result in an error message.)

   The Directory Server Plugin needs access to the configuration information you saved in the Console. To get this information, the Plugin communicates with the Directory Server Connector, through a server socket on this port. Additionally, the Plugin logs messages over this channel so the messages will go to the central log.

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.

Figure 5-4 Ready to Install Pane



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 is displayed because this step takes several minutes to complete.

NOTE If you did not close the Console before starting the installation, the following warning displays (Figure 5-5). Click Reset in the Console to reload the Connector's configuration settings.

Figure 5-5 Configuration Warning Dialog Box

When both steps are complete, an Installation Summary pane is displayed.

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 Windows**: Installation logs are written to the `%TEMP%` directory, which is usually a subdirectory of the `Local Settings` folder located under `C:\Documents and Settings\Administrator`

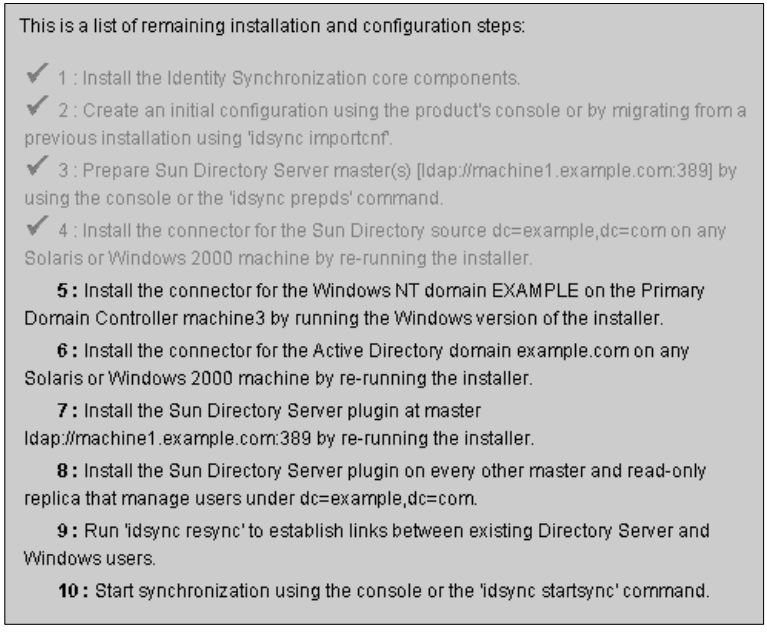
NOTE On some Windows systems (such as Windows 2000 Advanced Server), the `Local Settings` folder is a hidden folder.

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.

---

6. Click Next and the "To Do list" panel (Figure 5-6) displays to indicate which steps you have completed successfully and which steps remain.

Figure 5-6 To Do List



7. When you are done with the panel, click Finished.

After installing the Directory Server Connector, you can install other Connectors and / or Directory Server Plugins that you configured when you configured resources (Chapter 4):

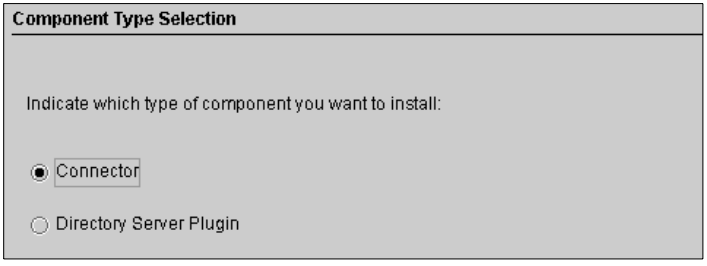
- Install additional Directory Server Connectors: Restart the installation program (using the instructions in “Running the Installation Program” on page 162) and then repeat Step 1 through Step 7.
- Install an Active Directory Connector: Go to “Installing an Active Directory Connector” on page 170.
- Install a Windows NT Connector: Go to “Installing the Windows NT Connector” on page 174.
- Install the Directory Server Plugin: Go to “Installing Directory Server Plugins” on page 175.

Installing an Active Directory Connector

After completing the steps described in “Running the Installation Program” on page 162, the Component Type Selection panel displays.

NOTE 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 or installing the Directory Server Plugin before you see the Connector Configuration pane (Figure 5-7).

Figure 5-7 Selecting the Connector



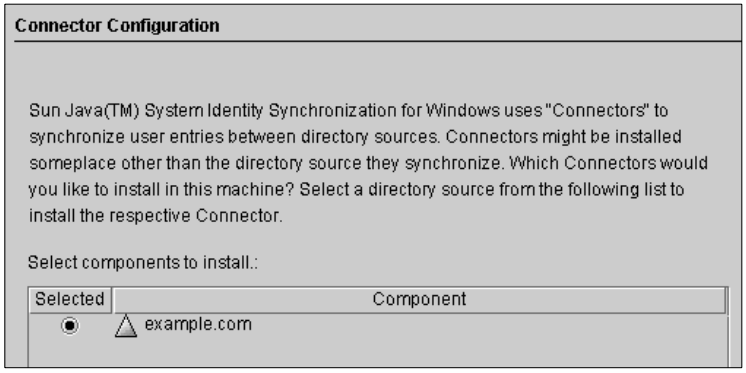
The component list contains only those Connector components that have not yet been installed. For example, if you already installed the Directory Server Connector (dc=example,dc=com in this case), it will not be listed.

To install an Active Directory Connector:

1. Enable the Connector button and click Next.

   The Connector Configuration panel displays (see Figure 5-8).

Figure 5-8 Selecting the Active Directory Connector

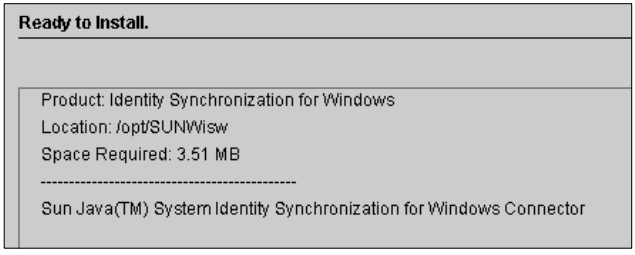


   The Select components to install list contains only those Connector components that have not yet been installed. For example, after you install the Directory Server Connector (dc=example,dc=com in this case), the program will remove the entry from this list pane.

2. Enable the button next to the Active Directory component and then click Next.

   The Ready to Install pane is displayed (Figure 5-9) to provide information about the Connector's installation location and how much disk space is required for the installation.

Figure 5-9 Ready to Install Pane



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.

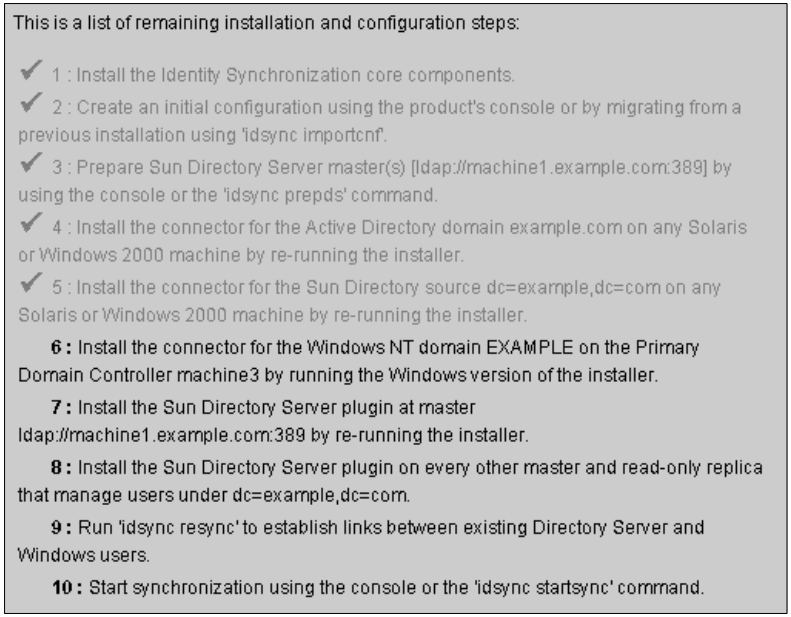
   - **On Solaris**: Installation logs are written to `/var/sadm/install/logs/`
   - **On Windows**: Installation logs are written to the `%TEMP%` directory, which is a subdirectory of the `Local Settings` folder located under `C:\Documents and Settings\Administrator`

NOTE On some Windows systems (such as Windows 2000 Advanced Server), the `Local Settings` folder is a hidden folder.

To view 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 and the “To Do list” panel is displayed (Figure 5-10) to indicate which steps you have completed successfully and which steps remain.

Figure 5-10 To Do List



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 and / or Directory Server Plugins that you configured when you configured resources (Chapter 4):

- Install additional Active Directory Connectors: Restart the installation program (see “Running the Installation Program” on page 162) and then repeat Step 1 through Step 6.
- Install a Windows NT Connector: Go to “Installing the Windows NT Connector” on page 174.
- Install additional Directory Server Connectors: Restart the installation program (using the instructions in “Running the Installation Program” on page 162) and then repeat Step 1 through Step 7.
- Install the Directory Server Plugin: Go to “Installing Directory Server Plugins” on page 175.

Installing the Windows NT Connector

NOTE You must install the Windows NT Connector on the Primary Domain Controller (PDC) of the domain you configured.

After completing the steps described in “Running the Installation Program” on page 162, the Connector Configuration panel displays.

To install a Windows NT Connector and the NT subcomponent(s):

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.)

   The Directory Server Plugin needs access to the configuration information you saved in the Console. To get this information, the Plugin communicates with the Windows NT Connector, through a server socket on this port. Additionally, the Plugin logs messages over this channel so the messages will go to the central log.
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 is displayed because this step takes several minutes to complete.

   NOTE If you did not close the Console before starting the installation, a warning displays (see Figure 5-5). 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 Finished to exit the installation program.

After installing the Windows NT Connector, you can install other Connectors and/or Directory Server Plugins that you configured when you configured resources (Chapter 4):

- Install additional Windows NT Connectors: Restart the installation program (see "Running the Installation Program" on page 162) and then repeat Step 1 through Step 6.
- Install an Directory Server Connector: Go to "Installing the Directory Server Connector" on page 165.
- Install an Active Directory Connector: Go to "Installing an Active Directory Connector" on page 170.
- Install the Directory Server Plugin: Go to "Installing Directory Server Plugins" on page 175.

Installing Directory Server Plugins

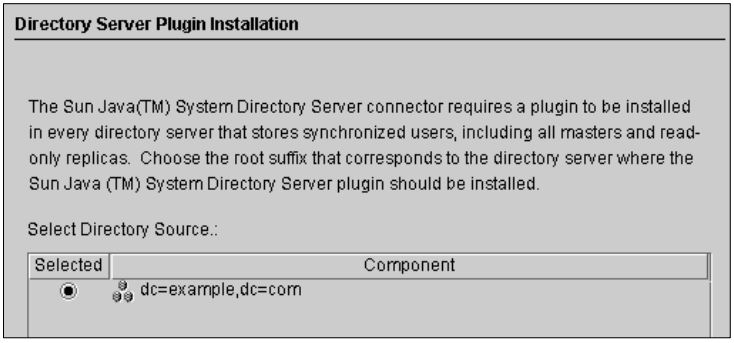
This section explains how to install the Identity Synchronization for Windows Directory Server Plugin.

NOTE You must install Directory Server Plugins on the same machine where you installed Directory Server.

If you are installing the Plugin on the same system as Core or any Connectors, the installation program will detect when Core or the Connectors have already been installed on the system. All additional components will be installed in the installation directory.

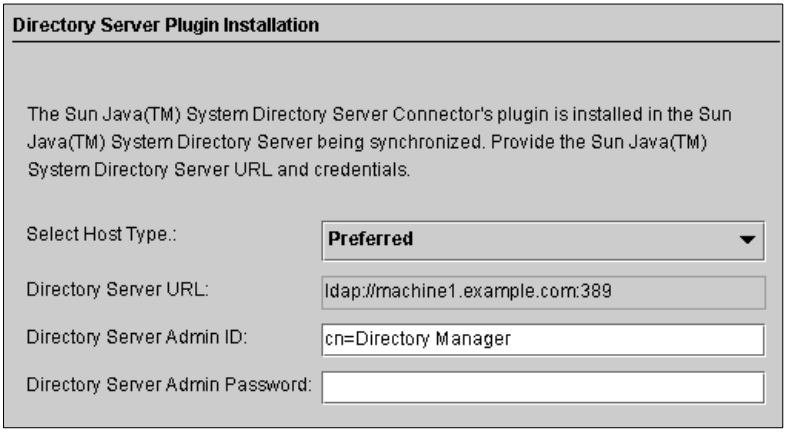
1. Complete the steps described in “Running the Installation Program” on page 162.

Figure 5-11 Selecting the Directory Server Plugin



2. When the Connector Configuration panel is displayed, enable the Directory Server Plugin (dc=example,dc=com) button and click Next.
3. Another Directory Server Plugin Installation pane is displayed (Figure 5-12).

Figure 5-12 Specifying the Directory Server URL and Credentials

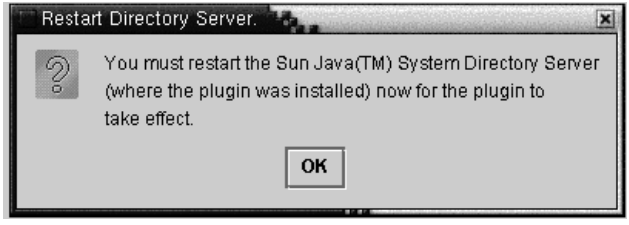


4. Select the appropriate Host Type from the drop-down list.
   - **Preferred**: Select this option if you are installing the Plugin on the preferred server.
   - **Secondary**: Select this option if you are installing the Plugin on a secondary server.
   - **Other**: Select this option if you are installing the Plugin on a machine that is not a preferred or secondary server.
5. Enter the URL where your Directory Server exists, if it is not a preferred or secondary host.
6. Enter the Directory Server administrator's name and password, and then click Next.

   The Ready to Install pane is displayed to provide information about the Plugin's installation location and how much disk space is required for the installation.
7. When you are ready, click the Install Now button.

   The Plugin 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 is displayed because this step takes several minutes to complete.
8. When both steps are complete, the following prompt is displayed. After reading the information, click OK to close the dialog box.

Figure 5-13 Restart Directory Server Prompt



9. Click the Details button if you want to review the installation log.

   - **On Solaris**: 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 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.

10. Click Finished to exit the installation program.

    The "To Do list" panel is displayed (similiar to Figure 5-10) to indicate which steps remain in the installation/configuration process.

    After installing the Directory Server Plugin, you can install other Connectors and/or Directory Server Plugins that you configured when you configured resources (Chapter 4):

    - Install additional Directory Server Plugins: Restart the installation program (see "Running the Installation Program" on page 162) and then repeat Step 2 through Step 9.
    - Because Identity Synchronization for Windows requires you to install the Plugin on every Directory Server in your deployment, you can continue running the Plugin installation program an unlimited number of times.
    - Install an Directory Server Connector: Go to "Installing the Directory Server Connector" on page 165.
    - Install an Active Directory Connector: Go to "Installing an Active Directory Connector" on page 170.
    - Install a Windows NT Connector: Go to "Installing the Windows NT Connector" on page 174.

11. If you have no other connectors or plugins to install, restart Directory Server.

Chapter 6

Synchronizing Existing Users

The Identity Synchronization for Windows command line utility provides the `idsync resync` subcommand to bootstrap deployments with existing users. 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 populations.

This chapter explains how to use the `idsync resync` subcommand to link and synchronize existing users 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 link and synchronize existing users.

For more information about the `idsync resync` subcommand, see Appendix A, "Using the Identity Synchronization for Windows Command Line Utilities."

Table 6-1 summarizes the post-installation steps to follow based on existing user populations:

Table 6-1 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	Using one of the following methods: • Run `idsync resync -f` <*filename*> to link and synchronize users from Active Directory and Directory Server. • Run `idsync resync -f` <*filename*> `-k` to link users only. • 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.

Using idsync resync

This section explains linking and 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:

- "Linking Users" on page 182
- "Resynchronizing Users" on page 181
- "idsync resync Arguments" on page 183
- "Checking Results in the Central Log" on page 186

Resynchronizing Users

NOTE Before starting synchronization for your deployment, verify that all existing users are synchronized between servers.

You can use the `idsync resync` command to link existing entries, create users, and synchronize user attributes in two directory sources. Specifically, you can use the `idsync resync` command to

- Populate an empty Directory Server with existing Active Directory or Windows NT SAM domain users
- Link all users and then synchronize all user entry attribute values (other than passwords) in two existing directory sources

NOTE If there are users that exist on Directory Server and Windows, you must run the `idsync resync -f` <*filename*> command to link and 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.

- Synchronize user entries when two directory sources become out of sync
- "Prime" the Active Directory and Windows NT SAM Connectors object cache database, which maintains a shadow copy of the Active Directory or Windows NT SAM user entries.

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).

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 running `idsync 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.

idsync resync Arguments

The `idsync resync` command accepts the following arguments:

Table 6-2 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, “LinkUsers XML Document Sample”)
`-k`	Creates links between unlinked users only (does not create users or modify existing users). You must use this argument in combination with the `-f` argument.
`-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 "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.*)

Table 6-2 idsync resync Usage *(Continued)*

Argument	Meaning
`-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 | NEW_LINKED_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 • **`NEW_LINKED_USERS`**: Forces on-demand password synchronization for all newly created or linked users See Table 6-3 for more information about how these options affect password validation.
`-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 6-3 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_LINKED_USERS`	No	Yes	Yes
`-i NEW_USERS`	No	No	Yes
No `-i` value	No	No	No

Table 6-4 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 6-4 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.
`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.

Table 6-4 idsync resync Usage Samples *(Continued)*

Arguments	Result
`idsync resync`	Displays a `resync` usage statement.
`idsync resync -f link.cfg -k -i NEW_LINKED_USERS`	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.

CAUTION 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.

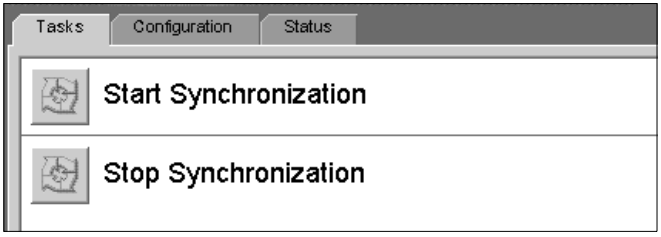
Starting and Stopping Synchronization

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 6-1):

Figure 6-1 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 “Using startsync” on page 330 and “Using stopsync” on page 331.

Starting and Stopping Services

Identity Synchronization for Windows and Message Queue are installed as *daemons* on Solaris 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 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.

Chapter 7

Migrating to Identity Synchronization for Windows 1 2004Q3

This chapter explains how to migrate your system from Sun Java System Identity Synchronization for Windows version 1.0 to version 1 2004Q3.

NOTE Identity Synchronization for Windows version 1.0 installed Message Queue for you, *but Identity Synchronization for Windows* 1 2004Q3 *does not*.

Refer to the Sun Java System Message Queue product documentation for installation instructions.

The information is organized into the following sections:

Overview

Migration from Identity Synchronization for Windows version 1.0 (or version 1.0 SP1) to 1 2004Q3 is accomplished in several major phases:

1. Preparing your Identity Synchronization for Windows version 1.0 (or 1.0 SP1) installation for Migration.
2. Uninstalling Identity Synchronization for Windows version 1.0 (or 1.0 SP1).
3. Installing or upgrading dependent products.
4. Installing Identity Synchronization for Windows 1 2004Q3 using the configuration and connector states you backed-up.

NOTE Install Identity Synchronization for Windows 1 2004Q3 on the same platform and architecture where you installed Identity Synchronization for Windows version 1.0 (or 1.0 SP1).

Before You Migrate

Before you start the migration process,

- Familiarize yourself with the new features and functionality provided in Sun Java System Identity Synchronization for Windows version 1 2004Q3.
- Read Chapter 2, “Preparing for Installation” for installation and configuration information you can use to plan your migration process.
- Document your version 1.0 deployment and configuration.
  Be sure to note any customizations you have made to the configuration.
- Schedule migration.
  Because the migration process requires at least four hours, you may want to schedule migration after normal business hours.

If users input password or attribute changes while you are migrating the system from version 1.0 to 1 2004Q3, Identity Synchronization for Windows will process these changes as follows:

- **For Active Directory**: Any password changes made on Active Directory during the migration process will be synchronized on-demand by the Directory Server Plugin after completing the migration process.
- **For Directory Server**: Any password changes made on Directory Server during migration will not be synchronized. However, you will be able to identify affected users in the Identity Synchronization for Windows 1 2004Q3 logs after completing the migration process. (See "Checking the Logs" on page 236.)
- **For Windows NT**: Any password changes made on NT during the migration process will not be synchronized.

  However, if you use the `forcepwchg` utility, you can identify affected users and force them to change passwords again. (See "Forcing Password Changes on Windows NT" on page 201 and "Checking the Logs" on page 236 for more information.)
- All other attribute changes made during migration (at any directory source) will be synchronized after you complete the migration process.

Preparing for Migration

You will use one or more the following utilities to migrate from version 1.0 to version 1 2004Q3:

- **`export10cnf`**: A stand-alone utility that enables you to create an export configuration file from your Identity Synchronization for Windows 1.0 configuration. (See "Exporting Your Version 1.0 Configuration" on page 192 for detailed information.)

  The exported XML document will contain the directory deployment's topology and enough information to configure the Identity Synchronization for Windows version 1 2004Q3 installation.
- **`checktopics`**: A utility that checks Message Queue synchronization topics in a 1.0 installation and determines if any undelivered messages remain on the queue.

Updates can remain in Message Queue after you stop 1.0 synchronization. You must verify that no updates exist in the Message Queue before you proceed with the migration. (See “Checking for Undelivered Messages” on page 199 for detailed information.)

- **`forcepwchg`**: A Windows NT tool that enables you to identify users who changed passwords during the migration process and force them to change passwords again when your version 1 2004Q3 system is ready. (Password changes made on Windows NT are not captured during the migration process.) (See “Forcing Password Changes on Windows NT” on page 201 for detailed information.)

NOTE These utilities facilitate the migration of Identity Synchronization for Windows version 1.0 to 1 2004Q3. The migration is performed in the same environment where Identity Synchronization for Windows 1.0 is deployed. Consequently, these utilities are available in the Solaris/SPARC and Windows packages only.

You will find the migration utilities in the installation `migration` directory — no additional installation steps are required.

Exporting Your Version 1.0 Configuration

You can use the `export10cnf` utility to export an existing version 1.0 configuration file to an XML file and then use the `idsync importcnf` command to quickly and accurately import the file into the 1 2004Q3 system before installing the connectors.

TIP Although it is possible to manually re-enter the 1.0 configuration using the Identity Synchronization for Windows console, you are strongly advised to use the `export10cnf` utility. If you decide not to use `export10cnf`, then you will not be able to preserve the state of your connectors.

Exporting your version 1.0 configuration provides the following benefits:

- You eliminate most of the initial configuration process to be performed from the management Console.
- You guarantee that the connector IDs assigned in version 1 2004Q3 will match the connector IDs used in version 1.0, which greatly simplifies the task of preserving the existing connector states that can be used directly in your version 1 2004Q3 deployment.

  (Basically, you back-up the `persist` and `etc` directories, and later restore them without having to worry about the underlying directory structure).

You will find the `export10cnf` utility in the installation `migration` directory — no additional installation steps are necessary.

Using the export10cnf Utility

To export an Identity Synchronization for Windows configuration to an XML file, execute `export10cnf` from the `migration` directory as follows:

- Open a Terminal window and type:

```
java -jar export10cnf.jar -h <hostname> -p <port> -D <bind DN>
-w <bind password> -s <rootsuffix> -q <configuration password> -Z
-P <cert-db-path> -m <secmod-db-path> -f <filename>
```

  For example,

```
java -jar export10cnf.jar -D "cn=dirmanager" -w - -q - -s
"dc=example,dc=com" -f exported-configuration
```

The `export10cnf` utility shares the same common arguments as the Identity Synchronization for Windows command-line utilities (see "Common Arguments" on page 312). The only `export10cnf` specific option is `-f <filename>`. If the operation is successful, then the utility will export the current configuration into the file specified in the argument of the -f option.

Inserting Clear-Text Passwords

The `export10cnf` utility does not export clear-text passwords from version 1.0 configurations (for security reasons). Instead, the utility inserts empty strings in `cleartextPassword` fields where appropriate. For example,

```
<Credentials
        userName="cn=iswservice,cn=users,dc=example,dc=com"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD
-->
```

You must enter a password manually — between the double quotes — for each and every `cleartextPassword` field in the exported configuration file *before you can import the file into Identity Synchronization for Windows 1 2004Q3*. (`importcnf` validation prevents you from importing a configuration file with empty password values.)

For example,

```
<Credentials
        userName="cn=iswservice,cn=users,dc=example,dc=com"
        cleartextPassword="mySecretPassword"/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE
FIELD -->
```

Sample Export Configuration File

Code Example 7-1 on page 195 contains a sample exported configuration file.

In this file,

- `ad-host.example.com` refers to the Active Directory domain controller.
- `ds-host.example.com` refers to the host running the Sun Java System Directory Server.

Code Example 7-1 Sample Export Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>

<ActiveConfiguration>
  <SunDirectorySource
    parent.attr="DirectorySource"
    onDemandSSLOption="true"
    maxConnections="5"
    displayName="dc=example,dc=com"
    resyncInterval="1000">
    <SynchronizationHost
      hostOrderOfSignificance="1"
      hostname="ds-host.example.com"
      port="389"
      portSSLOption="true"
      securePort="636">
      <Credentials
        userName="uid=PSWConnector,dc=example,dc=com"/>
    </SynchronizationHost>
<SyncScopeDefinitionSet
      index="0"
      location="ou=people,dc=example,dc=com"
      filter=""
      creationExpression="cn=%cn%,ou=people,dc=example,dc=com"
      sulid="SUL"/>
  </SunDirectorySource>
<ActiveDirectorySource
parent.attr="DirectorySource"
    displayName="example.com"
    resyncInterval="1000">
```

```
<SyncScopeDefinitionSet
      index="0"
      location="cn=users,dc=example,dc=com"
      filter=""
      creationExpression="cn=%cn%,cn=users,dc=example,dc=com"
      sulid="SUL"/>
</ActiveDirectorySource>
<ActiveDirectoryGlobals
    flowInboundCreates="true"
    flowInboundModifies="true"
    flowOutboundCreates="true"
flowOutboundModifies="true">
    <AttributeDescription
      parent.attr="CreationAttribute"
name="samaccountname"
syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeMap>
      <AttributeDescription
        parent.attr="WindowsAttribute"
        name="samaccountname"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
      <AttributeDescription
        parent.attr="SunAttribute"
        name="uid"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    </AttributeMap>
<AttributeDescription
      parent.attr="SignificantAttribute"
      name="description"
      syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeMap>
      <AttributeDescription
        parent.attr="SunAttribute"
        name="sn"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
        parent.attr="WindowsAttribute"
        name="sn"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    </AttributeMap>
```

```
<SynchronizationHost
      hostOrderOfSignificance="1"
      hostname="ad-host.example.com"
      port="389"
      portSSLOption="true"
      securePort="636">
      <Credentials
        userName="cn=iswservice,cn=users,dc=example,dc=com"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </SynchronizationHost>
<AttributeDescription
      parent.attr="CreationAttribute"
      name="cn"
      syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    <TopologyHost
      parent.attr="SchemaLocation"
hostname="ad-host.example.com"
      port="3268"
      portSSLOption="true"
      securePort="3269">
      <Credentials
        parent.attr="Credentials"
        userName="cn=iswservice,cn=users,dc=example,dc=com"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </TopologyHost>
    <TopologyHost
      parent.attr="HostsTopologyConfiguration"
      hostname="ad-host.example.com"
      port="3268"
      portSSLOption="true"
      securePort="3269">
<Credentials
        parent.attr="Credentials"
        userName="cn=iswservice,cn=users,dc=example,dc=com"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </TopologyHost>
<AttributeMap>
      <AttributeDescription
        parent.attr="SunAttribute"
        name="cn"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
```

```
 <AttributeDescription
        parent.attr="WindowsAttribute"
        name="cn"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    </AttributeMap>
<AttributeMap>
      <AttributeDescription
        parent.attr="SunAttribute"
        name="description"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
        parent.attr="WindowsAttribute"
name="description"
        syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    </AttributeMap>
</ActiveDirectoryGlobals>
<SunDirectoryGlobals
    userObjectClass="inetorgperson"
    flowInboundCreates="true"
    flowInboundModifies="true"
    flowOutboundCreates="true"
    flowOutboundModifies="true">
<TopologyHost
      parent.attr="SchemaLocation"
      hostname="ds-host.example.com"
      port="389"
      portSSLOption="false"
      securePort="636">
      <Credentials
        parent.attr="Credentials"
        userName="cn=directory manager"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </TopologyHost>
   <TopologyHost
      parent.attr="HostsTopologyConfiguration"
      hostname="ds-host.example.com"
      port="389"
      portSSLOption="false"
      securePort="636"><Credentials
        parent.attr="Credentials"
        userName="cn=directory manager"
        cleartextPassword=""/>
        <!-- INSERT PASSWORD BETWEEN THE DOUBLE QUOTES IN THE ABOVE FIELD -->
    </TopologyHost>
```

```
<AttributeDescription
      parent.attr="SignificantAttribute"
      name="description"
      syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
<AttributeDescription
      parent.attr="CreationAttribute"
      name="sn"
      syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
    <AttributeDescription
      parent.attr="CreationAttribute"
      name="cn"
      syntax="1.3.6.1.4.1.1466.115.121.1.15"/>
  </SunDirectoryGlobals>
</ActiveConfiguration>
```

At the completion of configuration export, `export10cnf` will report the result of the operation. If the operation fails, an appropriate error message is displayed with an error identifier.

Checking for Undelivered Messages

The Identity Synchronization for Windows 1.0 to 1 2004Q3 migration process minimizes system downtime by preserving the connectors' states in your existing deployment. However, these states reflect only the last change received and acknowledged by the Message Queue — so you will not know whether the message was actually delivered and applied to the destination connector.

This behavior does not cause problems as long as the Message Queue remains the same; however, you will lose any messages on the Message Queue during the migration process (when you install Message Queue 3.5 SP1).

You must verify that synchronization topics on the existing Message Queue do not have any undelivered messages before you proceed with the migration. The Identity Synchronization for Windows `checktopics` utility enables you to verify that all synchronization topics are empty (and the system is quiescent).

Using the checktopics Utility

The `checktopics` utility is delivered in the `migration` directory of the Solaris / SPARC and the Windows Identity Synchronization for Windows 1 2004Q3 package.

NOTE The only prerequisite for running `checktopics` is a suitable Java Virtual Machine (version 1.4.2_04 or later).

When you run the `checktopics` utility, it connects to the configuration directory, which contains information about Synchronization User Lists (SULs) and current synchronization topic names used in Message Queue. In addition, when you run `checktopics`, it queries the Message Queue to see how many outstanding messages remain on each active synchronization topic and then displays this information for you.

To execute the `checktopics` command line utility:

a. Open a Terminal window and **`cd`** to the `migration` directory.

b. From a command prompt, type the subcommand as follows:

```
java -jar checktopics.jar -h <hostname> -p <port> -D <bind_DN> -w
<bind_password> -s <root_suffix> -q <configuration_password> -Z
```

For example,

```
java -jar checktopics.jar -D "cn=directory manager" -w - -s
"dc=example,dc=com" -q -
```

NOTE

- For detailed information about the `checktopics` arguments, review “Common Arguments” on page 312.
- For more information about using `checktopics`, see “Checking for Undelivered Messages” on page 199.

After running checktopics, check your terminal for messages:

- If the operation was successful, a message is displayed to the terminal stating there are no outstanding messages in the logs.
- If the operation fails, an appropriate error message is displayed with an error identifier.

Clearing Messages

If any of the active synchronization topics have outstanding messages, you can use the following procedure to clear the messages:

1. Restart synchronization.
2. Wait until the messages are applied to the destination connector.
3. Stop synchronization.
4. Rerun checktopics.

Forcing Password Changes on Windows NT

On Windows NT, password changes are not monitored and new password values are not captured during the migration process. Consequently, there is no way to determine new password values after the migration is complete.

Instead of requiring all users to change passwords when you finish migrating to 1 2004Q3, you can use the forcepwchg command line utility to require a password change for all users who changed passwords during the migration process.

NOTE The forcepwchg utility is available in the Windows packages only.

You will find the forcepwchg utility in the Windows migration directory. You execute forcepwchg directly from that directory — no additional installation steps are necessary.

You must run forcepwchg on the Primary Domain Controller (PDC) host where the NT components (connector, Change Detector DLL, and Password Filter DLL) are installed — you cannot run forcepwchg remotely.

The forcepwchg utility also prints out the account names (one name per line) that it is trying to migrate. If an error occurs during the migration process, the error happened while migrating the user account that printed out last.

Migrating Your System

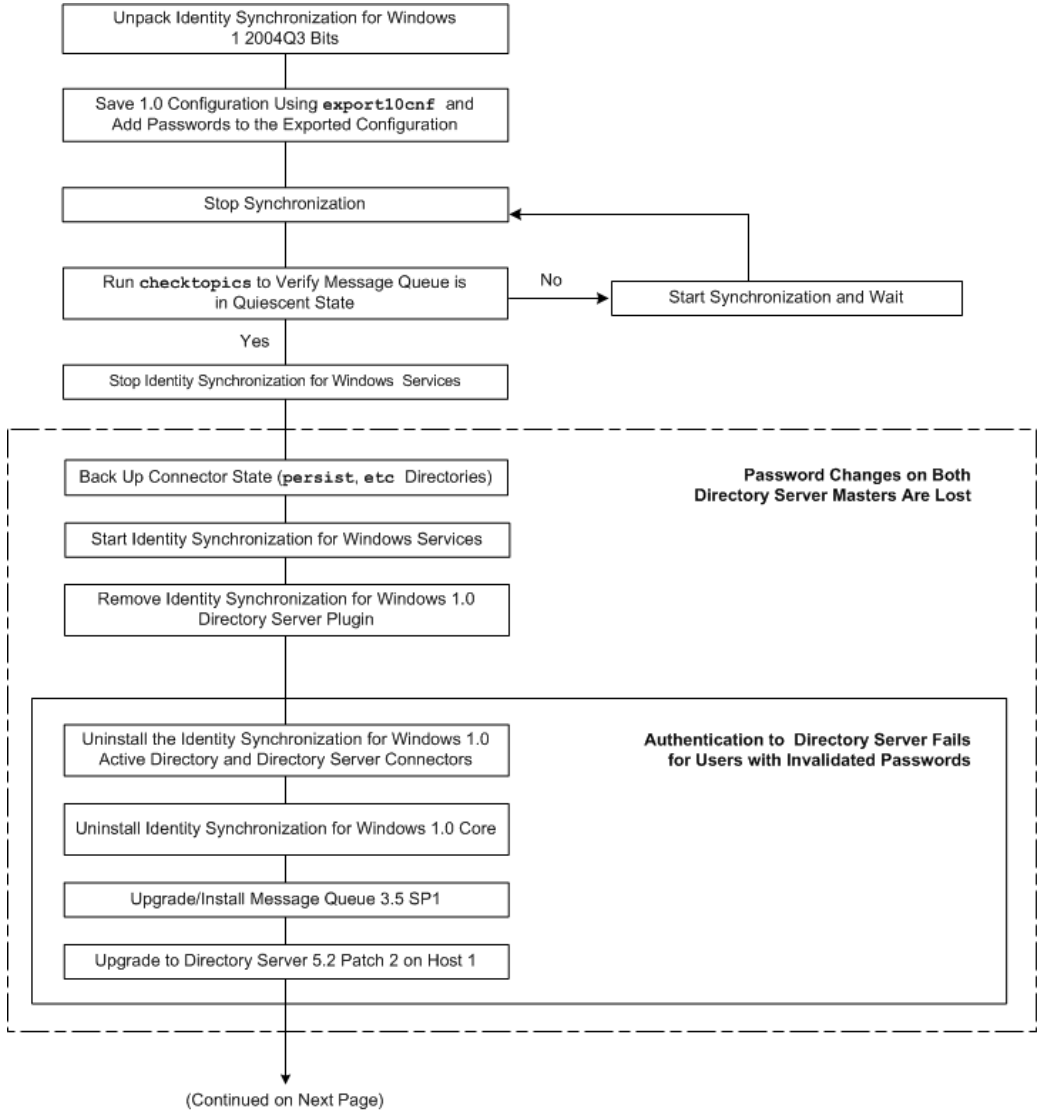
This section provides instructions for migrating a single-host deployment to version 1 2004Q3. In a single-host deployment, all Identity Synchronization for Windows components are installed on a single host (Windows 2000 Server, Solaris version 8 or 9, or SPARC), as follows:

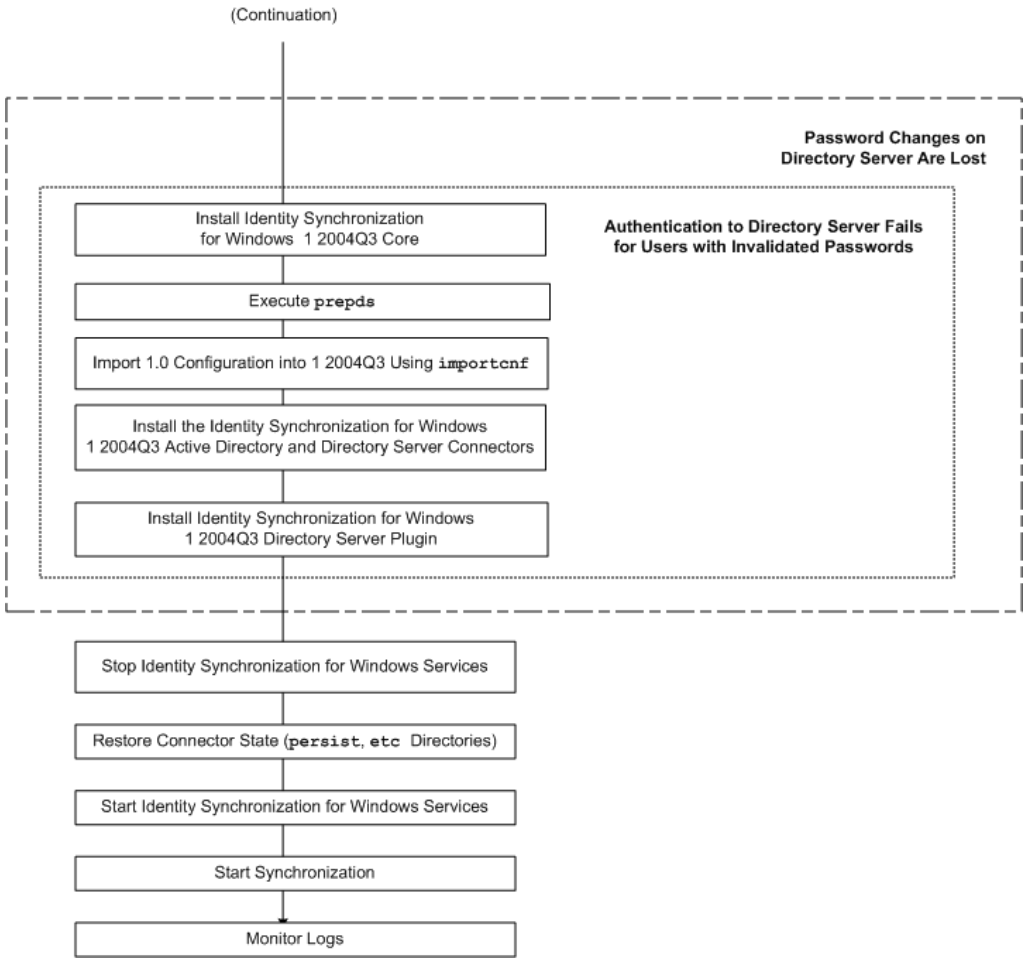
- Directory Server (one instance)
- Core (Message Queue, Central Logger, System Manager, and Console)
- Active Directory Connector
- Directory Server Connector
- Directory Server Plugin

NOTE If you are using Solaris as your installation host, then a Windows 2000 machine with Active Directory is required for synchronization purposes only. (No components would be installed on the Windows 2000 machine.)

The following figure illustrates the migration process and may serve as a checklist to supplement the migration instructions that follow.

Figure 7-1 Migrating a Single-Host Deployment




(Continuation)
Password Changes on Directory Server Are Lost
Install Identity Synchronization for Windows 1 2004Q3 Core
Authentication to Directory Server Fails for Users with Invalidated Passwords
Execute prepds
Import 1.0 Configuration into 1 2004Q3 Using importcnf
Install the Identity Synchronization for Windows 1 2004Q3 Active Directory and Directory Server Connectors
Install Identity Synchronization for Windows 1 2004Q3 Directory Server Plugin
Stop Identity Synchronization for Windows Services
Restore Connector State (persist, etc Directories)
Start Identity Synchronization for Windows Services
Start Synchronization
Monitor Logs

Preparing for Migration

Use the following procedure to prepare to migrate from Identity Synchronization for Windows version 1.0 to version 1 2004Q3:

1. From a command prompt:

   - **On Solaris or SPARC:** Type **`uncompress -c <filename> | tar xf -`**
   - **On Windows:** Type **`%JAVA_HOME%\bin\jar -xf <filename>`** (or use any zip archive program for Windows, such as WinZip®).

   After the binaries are unpacked, you will see the following subdirectories containing the necessary migration tools:

   - `installer/`
   - `lib/`
   - `migration/`

Solaris	Windows
export10cnf.jar	export10cnf.jar
—	forcepwchg.exe
checktopics.jar	checktopics.jar

2. Export your version 1.0 configuration settings to an XML file. From the `migration` directory, execute `export10cnf` as described in "Using the export10cnf Utility" on page 193. For example:

   ```
   java -jar export10cnf.jar -D "cn=directory manager" -w - -s
   "dc=example,dc=com" -q - -f export.cfg
   ```

3. Add passwords to the exported XML file.

   Enter a password between the double quotes for each `cleartextPassword` field in the exported configuration file (see "Inserting Clear-Text Passwords" on page 194).

4. Stop synchronization as described in "Starting and Stopping Synchronization" on page 187.

5. Verify that your system is in a quiescent state. From the `migration` directory, execute `checktopics` as described in "Using the checktopics Utility" on page 200.

For example:

```
java -jar checktopics.jar -D "cn=directory manager" -w - -s
"dc=example,dc=com" -q -
```

6. Stop Identity Synchronization for Windows services (daemons) as described in "Starting and Stopping Services" on page 188.

NOTE Do not stop the Sun ONE Message Queue service at this point.

7. *On Windows NT only* — Stop the Sun One NT ChangeDetector Service. You can stop the service from the command line by typing

   ```
   net stop "Sun One NT ChangeDetector Service"
   ```

8. *On Windows NT only* — Save the NT ChangeDetector Service counters as follows:

   a. Open the Registry Editor by executing `regedt32.exe`.

   b. Select the `HKEY_LOCAL_MACHINE` window.

   c. Navigate to the `SOFTWARE\Sun Microsystems\PSW\1.0` node.

   d. Save the following registry values:

   - `HighestChangeNumber`
   - `LastProcessedSecLogRecordNumber`
   - `LastProcessedSecLogTimeStamp`
   - `QueueSize`

9. Save the connector states by backing up the `persist` and `etc` directories from the existing 1.0 installation tree.

   - **On Solaris**: Type **`cd <serverroot>/isw-<hostname>`**
     `tar cf /var/tmp/connector-state.tar persist etc`
   - **On Windows**: Type **`cd <serverroot>\isw-<hostname>`**
     `zip -r C:\WINNT\Temp\connector-state.zip persist etc`
     `%JAVA_HOME%\bin\jar -cfM %TEMP%\connector-state.jar persist etc`
     (or use any zip archive program for Windows, such as WinZip)

10. Start the Identity Synchronization for Windows services (see page 188).

NOTE You do not have to start the Sun ONE Message Queue service because you never stopped it.

Uninstalling Identity Synchronization for Windows

NOTE The Identity Synchronization for Windows 1.0 uninstall program will remove the `SUNWjss` package if it is not registered for use by another application (other than Identity Synchronization for Windows 1.0). In particular, this situation may occur on Solaris machines if you installed a zip version of Directory Server 5.2.2, where the uninstall program removes the `jss3.jar` file from `/usr/share/lib/mps/secv1`.

If you encounter this situation as you migrate to Identity Synchronization for Windows 11 2004Q3, the installer will report that a required file is missing, and log the file name to the installation log. When this happens, you must re-install the required patches (see "Sun Java System Software Requirements" on page 56) and restart the installation process.

After completing the preparation steps, you are ready to begin uninstalling Identity Synchronization for Windows version 1.0 (or 1.0 SP1), as follows:

1. Uninstall the Directory Server Plugin manually, and restart each Directory Server where the Plugin was installed.
2. Execute the following steps on each Directory Server where the Plugin was installed:
   a. Remove the following entries from the Directory Server:

      ```
      cn=config,cn=pswsync,cn=plugins,cn=config
      cn=pswsync,cn=plugins,cn=config
      ```

      For example:

      `ldapdelete -D "cn=directory manager" -w - -p <port> -c cn=config, cn=pswsync,cn=plugins,cn=config cn=pswsync,cn=plugins,cn=config`

b. Restart the Directory Server.

- **On Solaris**: Type **<*serverroot*>/slapd-<*hostname*>/restart-slapd**
- **On Windows**: Type **<*serverroot*>\slapd-<*hostname*>\restart-slapd.bat**

c. Remove the Plugin binaries from the system.

- **On Solaris**: Type **rm <*serverroot*>/lib/psw-plugin.so**
  rm <*serverroot*>/lib/64/psw-plugin.so
- **On Windows**: Type **del <*serverroot*>\lib\psw-plugin.dll**

3. Change directory (**cd**) to **<*server_root*>\isw-<*hostname*>** and then use the Identity Synchronization for Windows 1.0 uninstallation program to uninstall the version 1.0/1.0 SP1 Connectors and Core components.

NOTE	You must always uninstall Connectors before uninstalling Core components.

- **On Solaris or SPARC**: Type **./runUninstaller.sh**
- **On Windows**: Type **\runUninstaller.bat**

4. Use the following steps to remove Identity Synchronization for Windows-related entries from the product registry file:

a. Back-up a copy of the file (located at):

- **On Solaris**: /var/sadm/install/productregistry
- **On Windows**: C:\WINNT\System32\productregistry

b. To remove the Identity Synchronization for Windows-related entries from the product registry file, follow the instructions provided in Step 6 of the "Manually Uninstalling 1.0 Core and Instances from Solaris."

5. *On Windows only* — After uninstalling Core, restart your machine.

NOTE	If the uninstall fails for any reason, you may have to manually uninstall the Identity Synchronization for Windows components. Instructions are provided in "What to Do if the 1.0 Uninstallation Fails" on page 212.

6. *On Windows only* — Verify that Identity Synchronization for Windows is not running. If necessary, you can stop the service from the command line by typing

   `net stop "Sun ONE Identity Synchronization for Windows"`

   If this service continues running after uninstallation is completed, it causes a sharing violation that will prevent you from deleting the instance directory.

7. Remove the Identity Synchronization for Windows instance directory (`isw-<`*`hostname`*`>`).

Installing or Upgrading the Dependent Products

Use the following steps to upgrade the Java Runtime Environment, install Message Queue, and upgrade Directory Server:

1. Upgrade the Java 2 Runtime Environment (or Java 2 SDK) on each host (except on Windows NT) where Identity Synchronization for Windows components are installed. (The minimum required version is 1.4.2_04.)

   - **Java 2 SDK**: `http://java.sun.com/j2se/1.4.2/install.html`
   - **Java 2 Runtime Environment**: `http://java.sun.com/j2se/1.4.2/jre/install.html`

2. Install Message Queue 3.5 SP1 using instructions provided in the *Sun Java System Message Queue 3.5 SP1 Installation Guide.*

3. Upgrade Directory Server to version 5.2 SP2 using instructions provided in the *Sun Java System Directory Server 5 2004Q2 Installation and Migration Guide*, which is available from the following location:

   `http://docs.sun.com/db/coll/DirectoryServer_04q2`

   The Directory Server upgrade will preserve your current Directory Server configuration and database.

Installing Identity Synchronization for Windows 1 2004Q3

Use the following steps to install the Identity Synchronization for Windows 1 2004Q3 components:

1. Install Identity Synchronization for Windows 1 2004Q3 Core. (see "Installing Core" on page 89)
2. Execute `idsync prepds` against Directory Server as follows to update the schema.
   - **On Solaris**: Type **`cd /opt/SUNWisw/bin`**
     Then type: **`idsync prepds <`*`arguments`*`>`**
   - **On Windows**: Type **`cd \<`*`serverroot`*`>\isw-<`*`hostname`*`>\bin`**
     Then type: **`idsync prepds <`*`arguments`*`>`**

   For more information about `idsync prepds`, see Appendix A, "Using the Identity Synchronization for Windows Command Line Utilities."
3. Import your version 1.0 configuration XML file by typing

   `idsync importcnf <`*`arguments`*`>`

   NOTE If the program detects errors in your input configuration file, an error will result. Identity Synchronization for Windows will abort the `importcnf` process and provide necessary information to correct the mistakes.

   For more information about using `idsync importcnf`, see "Using importcnf" in Appendix A.
4. Install the Identity Synchronization for Windows 1 2004Q3 Connectors (see "Installing Connectors" on page 164).
5. Install Identity Synchronization for Windows 1 2004Q3 Directory Server Plugin ("Installing Directory Server Plugins" on page 175).
6. Stop Identity Synchronization for Windows services (daemons) as described in "Starting and Stopping Services" on page 188.
7. *On Windows NT only* — Stop the Sun Java™ System NT Change Detector service. You can stop the service from the command line by typing

   `net stop "Sun Java(TM) System NT Change Detector"`

8. *On Windows NT only* — Restore the NT ChangeDetector Service counters:

   a. Open the Registry Editor by executing `regedt32.exe`.

   b. Select the `HKEY_LOCAL_MACHINE` window.

   c. Navigate to the `SOFTWARE\Sun Microsystems\Sun Java(TM) System Identity Synchronization for Windows\1.1` node.

   d. Double-click on each of the following entries to restore their values (which you saved prior to uninstalling version 1.0):

      - `HighestChangeNumber`
      - `LastProcessedSecLogRecordNumber`
      - `LastProcessedSecLogTimeStamp`
      - `QueueSize`

9. *On Windows NT only* — Start the Sun Java™ System NT Change Detector service. You can start the service from the command line by typing

   `net start "Sun Java(TM) System NT Change Detector"`

10. Remove the 1 2004Q3 `persist` and `etc` directories (and all their contents) from the instance directory and restore the version 1.0 (or 1.0 SP1) `persist` and `etc` directories you backed up in "Preparing for Migration" on page 205.

    - **On Solaris**: Type

      ```
      cd /var/opt/SUNWisw
      rm -rf etc persist
      tar xf /var/tmp/connector-state.tar
      ```

    - **On Windows**: Type

      ```
      cd <serverroot>\isw-<hostname>
      rd /s etc persist
      %JAVA_HOME%\bin\jar -xf %TEMP%\connector-state.jar
      ```

      (or use any zip archive program for Windows, such as WinZip)

11. Start Identity Synchronization for Windows service (see page 188).

12. Start synchronization as described in "Starting and Stopping Synchronization" on page 187.

13. Check the central audit log to verify there are no warning messages.

NOTE If you customized your version 1.0 log settings, you must manually apply those customizations to your version 1 2004Q3 installation. Use the Identity Synchronization for Windows Console to configure your version 1 2004Q3 log settings.

What to Do if the 1.0 Uninstallation Fails

If the version 1 2004Q3 installation program finds remnants of the version 1.0 system, then the 1 2004Q3 installation will fail. Consequently, you should verify that all of the 1.0 components are completely removed from the system prior to installing version 1 2004Q3.

If the uninstallation program does not uninstall all of the version 1.0/1.0 SP1 components, you must manually clean up the Identity Synchronization for Windows product registry and Solaris packages.

Detailed instructions for uninstalling Identity Synchronization for Windows version 1.0 manually are provided in the following three sections:

- “Manually Uninstalling 1.0 Core and Instances from Solaris” on page 213
- “Manually Uninstalling 1.0 Core and Instances from Windows 2000” on page 219
- “Manually Uninstalling a 1.0 Instance from Windows NT” on page 225

NOTE The instructions provided in this section are for uninstalling Identity Synchronization for Windows *version 1.0* only.

Do not use the manual uninstallation procedures provided in the following sections unless the Identity Synchronization for Windows uninstallation program fails.

Manually Uninstalling 1.0 Core and Instances from Solaris

Use the instructions provided in this section to manually uninstall Core from a Solaris machine.

NOTE In this section, Identity Synchronization for Windows locations are described in the following manner:

<*serverroot*>/isw-<*hostname*>

Where <*serverroot*> represents the parent directory of the Identity Synchronization for Windows installation location.

For example, if you installed Identity Synchronization for Windows in /var/Sun/mps/isw-<*example*>, the <*serverroot*> would be /var/Sun/mps.

---

1. Stop all Identity Synchronization for Windows Java processes by typing **/etc/init.d/isw stop** into a terminal window.

   If the preceding command does not stop all of the Java processes, type the following:

   /usr/ucb/ps -gauxwww | grep java

   kill -s SIGTERM <*process IDs from preceding command*>

2. Stop Message Queue as follows:

   a. At the prompt, type the following command to stop the Message Queue broker:

      /etc/init.d/imq stop

   b. To stop any remaining imq processes, type:

      *** ps -ef | grep imqbroker**

      *** kill -s SIGTERM <*process IDs from preceding command*>**

c. Use one of the following methods to uninstall the broker packages and directories:

- Use the Message Queue broker uninstall script (located in the Identity Synchronization for Windows instance directory on the host where you installed Core) to uninstall the broker. Type the following:

  /<*serverroot*>/isw-<*hostname*>/imq_uninstall

- Manually uninstall the packages and directories as follows:

  Use the `pkgrm`: command to remove these packages:

`SUNWaclg`	`SUNWiqum`	`SUNWiqjx`
`SUNWiqlen`	`SUNWxsrt`	`SUNWiqu`
`SUNWjaf`	`SUNWiqfs`	`SUNWjhrt`
`SUNWiqdoc`	`SUNWiquc`	`SUNWiqsup`
`SUNWiqr`	`SUNWjmail`	

  Use the `rm -rf` command to remove these directories:

```
rm -rf /etc/imq
rm -rf /var/imq
rm -rf /usr/bin/imq*
```

3. To remove the Identity Synchronization for Windows 1.0 Solaris packages, run `pkgrm` <*packageName*> for each of the packages listed in Table 7-1. (For example, **`pkgrm SUNWidscm SUNWidscn SUNWidscr SUNWidsct SUNWidsoc`**)

Table 7-1 Solaris Packages to Remove

Package Name	Description
SUNWidscm	Sun ONE Directory Server Identity Synchronization package for Core components and Connectors.
SUNWidscn	Sun ONE Directory Server Identity Synchronization package for Console help files.
SUNWidscr	Sun ONE Directory Server Identity Synchronization package for Core Components.
SUNWidsct	Sun ONE Directory Server Identity Synchronization package for Connectors.
SUNWidsoc	Sun ONE Directory Server Identity Synchronization package for Object Cache.

To verify that all of the packages were removed, type the following:

```
pkginfo | grep -i "Identity Synchronization"
```

NOTE Run the `pkgrm <packageName>` command again if there are still existing packages due to dependencies.

4. Remove Director Server Plugin as follows:

   a. Open the Directory Server Console and select the Configuration tab.

   b. In the left pane, expand the Plugins node and select the pswsync node.

   c. In the right pane, uncheck the Enable plug-in check box.

   d. Click Save to save your changes.

   e. From the Directory Server Console, locate and remove the following entry from the Configuration Directory:

      `cn=pswsync,cn=plugins,cn=config`

   f. Stop Directory Server.

   g. To remove the Plugin binary, type

      `rm -f /<serverroot>/lib/psw-plugin.so`

   h. Restart Directory Server.

5. Back-up (copy and rename) the current `productregistry` file located in `/var/sadm/install/productregistry`.

6. Manually edit the `productregistry` file in `/var/sadm/install/` to remove the following entries (*if present*):

NOTE

- For best results, use an XML editor. Alternatively, you can use a standard text editor.
- Some of the following components may not be included in your file.
- You must delete the beginning tag (`<compid>`), ending tag (`<\compid>`), and all contents in-between both tags). Ellipses are used in the following list to represent any additional text and / or tags that are included as part of these tags. (See the example on page 217.)

- `<compid>Identity Synchronization for Windows . . . </compid>`
- `<compid>Core . . . </compid>`
- `<compid>uninstaller . . . </compid>`
- `<compid>wpsyncwatchdog . . . </compid>`
- `<compid>setenv . . . </compid>`
- `<compid>Create DIT . . . </compid>`
- `<compid>Extend Schema . . . </compid>`
- `<compid>resources . . . </compid>`
- `<compid>CoreComponents . . . </compid>`
- `<compid>Connector . . . </compid>`
- `<compid>DSConnector . . . </compid>`
- `<compid>Directory Server Plugin . . . </compid>`
- `<compid>DSSubcomponents . . . </compid>`
- `<compid>ObjectCache . . . </compid>`
- `<compid>ObjectCacheDLLs . . . </compid>`
- `<compid>SUNWidscr . . . </compid>`

- <compid>SUNWidscm . . . </compid>
- <compid>SUNWidsct . . . </compid>
- <compid>SUNWidscn . . . </compid>
- <compid>SUNWidsoc . . . </compid>
- <compid>ADConnector . . . </compid>

The following is a example <compid> tag. Remove <compid>, </compid>, and all text and tags in-between.

```
<compid>Identity Synchronization for Windows
    <compversion>1.0
          <uniquename>Identity Synchronization for Windows</uniquename>
           <compinstance>1
                    <children>
                        <compref>ADConnector
                            <instance>1
                                <version>1.0</version>
                            </instance>
                        </compref>
                        <compref>DSSubcomponents
                        . . .

        </compinstance>
    </compversion>
</compid>
```

7. Remove the following Identity Synchronization for Windows directories and files:

   a. From the installation location, type

      rm -rf /<*serverroot*>/isw-<*hostname*>

   b. Remove the bootstrap files by typing

      rm -rf /etc/init.d/isw

8. Clean up the configuration directory as follows:

   a. Run the following `ldapsearch` command against the configuration directory where Identity Synchronization for Windows Core is installed to locate the Identity Synchronization for Windows Console subtree:

      ```
      ldapsearch -D "cn=directory manager" -w <my_password> -b
      o=netscaperoot "(nsnickname=isw)" dn
      ```

      NOTE `ldapsearch` is located in Directory Server's `<serverroot>/shared/bin/ldapsearch`
      For example, `/var/Sun/mps/shared/bin/ldapsearch`

      The resulting entry should be similar to the following (note that the entry will always end with *o=NetscapeRoot*):

      ```
      "cn=Sun ONE Identity Synchronization for Windows,cn=server
      group, cn=myhost.mydomain.com,ou=mydomain.com,o=NetscapeRoot"
      ```

   b. Use the Directory Server Console to remove the Identity Synchronization for Windows Console subtree and all subtrees below it.

9. Clean up the Identity Synchronization for Windows configuration registry as follows:

   a. Run the following `ldapsearch` command to locate the Identity Synchronization for Windows configuration registry in Directory Server:

      ```
      ldapsearch -D "cn=directory manager" -w <my_password> -b
      "dc=my,dc=domain"
      "(&(objectclass=iplanetservice)(ou=IdentitySynchronization))" dn
      ```

      The resulting entry should be similar to the following:

      ```
      "ou=IdentitySynchronization,ou=Services,dc=my,dc=domain"
      ```

   b. Use the Directory Server Console to remove the Identity Synchronization for Windows configuration registry and all subtrees below it.

10. Clean up all other Console-related files as follows:

    a. Remove all Console jar files by typing:

    `rm -rf <serverroot>/java/jars/isw*`
    For example, `/var/Sun/mps/java/jars/isw*`

    b. Remove all Console servlet jar files by typing:

    `rm -rf <serverroot>/bin/isw/`
    For example, `/var/Sun/mps/bin/isw/`

Manually Uninstalling 1.0 Core and Instances from Windows 2000

Use the instructions provided in this section to manually uninstall Core from a Windows 2000 machine.

NOTE In this section, Identity Synchronization for Windows locations are described in the following manner:

<*serverroot*>\isw-<*hostname*>

Where <*serverroot*> represents the parent directory of the Identity Synchronization for Windows installation location.

For example, if you installed Identity Synchronization for Windows in `C:\Program Files\Sun\mps\isw-`*`example`*, the <*serverroot*> would be `C:\Program Files\Sun\mps`.

1. Stop all Identity Synchronization for Windows Java processes using one of the following methods:

    - Select Start > Settings > Control Panel > Administrative Tools > Services to open the Services window. In the right pane, right-click on Sun ONE Identity Synchronization for Windows and select Stop.
    - Open a Command Prompt window and type the following command:

    `net stop "Sun ONE Identity Synchronization for Windows"`

- If the preceding methods do not work, you can use the following steps to stop the Java processes manually:

  I. Open the Services window, right-click on Sun ONE Identity Synchronization for Windows, and select Properties.

  II. From the General tab in the Properties window, select Manual from the Startup type drop-down list.

NOTE Although you can view Java processes (such as `pswwatchdog.exe`) from the Windows Task Manager, you cannot determine which processes are specifically related to Identity Synchronization for Windows. For this reason, do not stop processes from the Windows Task Manager.

---

2. Stop the Message Queue (for a Core uninstallation only) using one of the following methods:

- In the Services window, right-click on iMQ Broker in the right pane and select Stop.
- Open a Command Prompt window and type the following command:

  `net stop "iMQ Broker"`

- If the preceding methods do not work, you can use the following steps to stop Message Queue manually:

  I. Open the Services window, right-click on iMQ Broker and select Properties.

  II. From the General tab in the Properties window, select Manual from the Startup type drop-down list.

3. Remove the Directory Server Plugin as follows:

a. Open the Directory Server Console and select the Configuration tab.

b. In the left pane, expand the Plugins node and select the pswsync node.

c. In the right pane, uncheck the Enable plug-in check box.

d. Click Save to save your changes.

e. From the Console, locate and remove the following entry from the Configuration Directory:

`cn=pswsync,cn=plugins,cn=config`

f. Stop Directory Server, using one of the following methods:

- In the Services window, right-click on Sun ONE Directory Server 5.2 in the right pane and select Stop.
- Open a Command Prompt window and type the following command:

  net stop slapd-<*myhostname*>

g. Open the Windows Explorer to locate and remove the Plugin binary:

<*serverroot*>\lib\psw-plugin.so

h. Restart Directory Server.

4. Open a Command Prompt window and type **regedit** to open the Registry Editor window.

Important – Back up your current registry file before proceeding to Step 5.

a. In the Registry Editor, select the top node (My Computer) in the left pane.

b. Select Registry > Export Registry File from the menu bar.

c. When the Export Registry File dialog box is displayed, specify a name for the file and select a location in which to save the backup registry.

5. In the Registry Editor, select Edit > Delete from the menu bar and remove the following Identity Synchronization for Windows keys from the Windows Registry:

- All entries under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Identity Synchronization for Windows
- All CurrentControlSet and ControlSet (such as ControlSet001, ControlSet002, and so forth) entries under HKEY_LOCAL_MACHINE\SYSTEM*, which includes the following entries (if they exist):
  - ...\Control\Session Manager\Environment\<*isw-installation directory*>
  - ...\Services\Eventlog\Application\Sun ONE Identity Synchronization for Windows
  - ...\Services\Sun ONE Identity Synchronization for Windows
  - ...\Services\iMQBroker

6. Back-up (copy and rename) the current productregistry file located in C:\WINNT\system32.

7. Edit the `C:\WINNT\system32 \productregistry` file to remove the following tags:

NOTE

- For best results, use an XML editor. Alternatively, you can use a standard text editor.
- Some of the following components may not be included in your file.
- You must delete the beginning tag (`<compid>`), ending tag (`<\compid>`), and all contents in-between both tags). Ellipses are used in the following list to represent any additional text and/or tags that are included as part of these tags. (See the example page 223.)

- `<compid>Identity Synchronization for Windows . . . </compid>`
- `<compid>Core . . . </compid>`
- `<compid>unistaller . . . </compid>`
- `<compid>wpsyncwatchdog . . . </compid>`
- `<compid>setenv . . . </compid>`
- `<compid>Create DIT . . . </compid>`
- `<compid>Extend Schema . . . </compid>`
- `<compid>resources . . . </compid>`
- `<compid>CoreComponents . . . </compid>`
- `<compid>Connector . . . </compid>`
- `<compid>DSConnector . . . </compid>`
- `<compid>Directory Server Plugin . . . </compid>`
- `<compid>DSSubcomponents . . . </compid>`
- `<compid>ObjectCache . . . </compid>`
- `<compid>ObjectCacheDLLs . . . </compid>`
- `<compid>ADConnector . . . </compid>`

The following is a <compid> tag sample. Remove <compid>, </compid>, and all text and tags in-between.

```
<compid>Identity Synchronization for Windows
    <compversion>1.0
          <uniquename>Identity Synchronization for Windows</uniquename>
           <compinstance>1
                    <children>
                        <compref>ADConnector
                            <instance>1
                                <version>1.0</version>
                            </instance>
                        </compref>
                        <compref>DSSubcomponents
                        . . .

        </compinstance>
    </compversion>
</compid>
```

8. Remove the Identity Synchronization for Windows installation folder located at <*serverroot*>\isw-<*hostname*>.

   For example, C:\Program Files\Sun\mps\isw-example

9. Clean up the configuration directory as follows:

   a. From a Command Prompt window, run the ldapsearch command against the configuration directory where Identity Synchronization for Windows Core is installed to locate the Identity Synchronization for Windows Console subtree.

   NOTE ldapsearch is located in <*serverroot*>\shared\bin\ldapsearch.

   For example, C:\Program Files\Sun\mps\shared\bin\ldapsearch

```
ldapsearch -D "cn=directory manager" -w <my_password> -b
o=netscaperoot "(nsnickname=isw)" dn
```

The resulting entry should be similar to the following (note that the entry will always end with *o=NetscapeRoot*):

```
"cn=Sun ONE Identity Synchronization for Windows,cn=server
group, cn=myhost.mydomain.com,ou=mydomain.com,o=NetscapeRoot"
```

b. Use the Directory Server Console to remove the Identity Synchronization for Windows Console subtree you found and all subtrees below it.

10. Clean up the Identity Synchronization for Windows configuration directory (*also know as the configuration registry*) as follows:

a. From a Command Prompt window, run the following `ldapsearch` command to locate the Identity Synchronization for Windows configuration directory in Directory Server:

```
ldapsearch -D "cn=directory manager" -w <my_password> -b
"dc=my,dc=domain"
"(&(objectclass=iplanetservice)(ou=IdentitySynchronization))" dn
```

The resulting entry should be similar to the following:

```
"ou=IdentitySynchronization,ou=Services,dc=my,dc=domain"
```

b. Use the Directory Server Console to remove the configuration directory subtree you found, including all subtrees below it.

11. Clean up all other Console-related files as follows:

a. Remove all Console jar files located in <*serverroot*>`\java\jars\isw*`
For example, `C:\Program Files\Sun\mps\java\jars\isw*`

b. Remove all Console servlet jar files located in
`\`<*directory_server_install_root*>`\bin\isw\`
For example, `C:\SunOne\Servers\bin\isw\`

12. Restart your machine for all changes to take effect.

Manually Uninstalling a 1.0 Instance from Windows NT

Use the instructions provided in this section to manually uninstall an instance from a Windows NT machine.

NOTE In this section, Identity Synchronization for Windows locations are described in the following manner:

<*serverroot*>\isw-<*hostname*>

Where <*serverroot*> represents the parent directory of the Identity Synchronization for Windows installation location. For example, if you installed Identity Synchronization for Windows in C:\Program Files\Sun\mps\isw-*example*, the <***serverroot***> would be C:\Program Files\Sun\mps.

1. Stop all Identity Synchronization for Windows Java processes (Core and instance installations) using one of the following methods:
   - Select Start > Settings > Control Panel > Administrative Tools > Services to open the Services window. In the right pane, right-click on Sun ONE Identity Synchronization for Windows and select Stop.
   - Open a Command Prompt window and type the following command:

     net stop "Sun ONE Identity Synchronization for Windows"
   - If the preceding methods do not work, use the following steps to stop the Java processes manually:

     I. Open the Services window, right-click on Sun ONE Identity Synchronization for Windows, and select Properties.

     II. From the General tab in the Properties window, select Manual from the Startup type drop-down list.

     NOTE Although you can view Java processes (such as pswwatchdog.exe) from the Windows Task Manager, you cannot determine which processes are specifically related to Identity Synchronization for Windows. For this reason, do not stop processes from the Windows Task Manager.

2. Stop the Change Detector service using one of the following methods:
   - In the Services window, right-click on Sun ONE NT Change Detector Service in the right pane and select Stop.
   - Open a Command Prompt window and type the following command:

     `net stop "Sun ONE NT Change Detector Service"`
   - If the preceding methods do not work, use the following steps to stop the Change Detector Service manually:

     I. Open the Services window, right-click on Change Detector Service and select Properties.

     II. From the General tab in the Properties window, select Manual from the Startup type drop-down list.
3. Restart your Windows NT computer.
4. You must remove Identity Synchronization for Windows registry keys. Open a Command Prompt window and type **`regedt32`** to open the Registry Editor window.

CAUTION *Do not* use `regedit` because the program does not allow you to edit multi-value strings.

Be sure to back up your current Windows registry file *before* proceeding to Step 5.

---

   a. In the Registry Editor, select the top node (My Computer) in the left pane.
   b. Select Registry > Export Registry File from the menu bar.
   c. When the Export Registry File dialog box is displayed, specify a name for the file and select a location in which to save the backup registry.

5. In the Registry Editor, select Edit > Delete from the menu bar and remove the following Identity Synchronization for Windows keys from the Registry:
   - All entries under `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Identity Synchronization for Windows`
   - All `CurrentControlSet` and `ControlSet` (such as `ControlSet001`, `ControlSet002`, and so forth) entries under `HKEY_LOCAL_MACHINE\SYSTEM\*`, which includes the following entries (if they exist):
     - `...\Control\Session Manager\Environment\<`*isw-installation directory*`>`
     - `...\Services\Eventlog\Application\Sun ONE Identity Synchronization for Windows`
     - `...\Services\Sun ONE Identity Synchronization for Windows`
     - `...\Services\iMQBroker`
   - The `HKEY_LOCAL_MACHINE\SOFTWARE\Sun Microsystems\PSW`
6. Use **`regedt32`** (*do not use* `regedit`) to modify (***do not delete***) the following registry key:
   a. Select the registry key entry in the left pane:

      `HKEY_LOCAL_MACHINE\SYSTEM\\CurrentControlSet\\CONTROL\\LSA`

      The registry value type must be `REG_MULTI_SZ`.

   b. In the right pane, right-click on the Notification Packages value and select Modify.

   c. Change the `PASSFLT` value to `FPNWCLNT`.
7. Back-up (copy and rename) the current `productregistry` file located in `C:\WINNT\system32`.

8. Edit the `C:\WINNT\system32` `productregistry` file to remove the following tags:

NOTE

- For best results, use an XML editor. Alternatively, you can use a standard text editor.
- Some of the following components may not be included in your file.
- You must delete the beginning tag (`<compid>`), ending tag (`<\compid>`), and all contents in-between both tags). Ellipses are used in the following list to represent any additional text and/or tags that are included as part of these tags. (See the example on page 223.)

- `<compid>Identity Synchronization for Windows . . . </compid>`
- `<compid>Core . . . </compid>`
- `<compid>uninstaller . . . </compid>`
- `<compid>wpsyncwatchdog . . . </compid>`
- `<compid>setenv . . . </compid>`
- `<compid>Create DIT . . . </compid>`
- `<compid>Extend Schema . . . </compid>`
- `<compid>resources . . . </compid>`
- `<compid>CoreComponents . . . </compid>`
- `<compid>Connector . . . </compid>`
- `<compid>DSConnector . . . </compid>`
- `<compid>Directory Server Plugin . . . </compid>`
- `<compid>DSSubcomponents . . . </compid>`
- `<compid>ObjectCache . . . </compid>`
- `<compid>ObjectCacheDLLs . . . </compid>`
- `<compid>ADConnector . . . </compid>`

The following is a example <compid> tag. Remove <compid>, </compid>, and all text and tags in-between.

```
<compid>Identity Synchronization for Windows
    <compversion>1.0
       <uniquename>Identity Synchronization for Windows</uniquename>
           <compinstance>1
                    <children>
                        <compref>ADConnector
                            <instance>1
                                <version>1.0</version>
                            </instance>
                        </compref>
                        <compref>DSSubcomponents
                        . . .

         </compinstance>
    </compversion>
</compid>
```

9. Remove the Identity Synchronization for Windows installation folder located at *<**serverroot**>*\isw-*<hostname>*.

   For example, C:\Program Files\Sun\mps\isw-example

   NOTE You must edit the Windows registry as described in Step 8 before proceeding to Step 10.

10. Remove the Password Filter DLL.

    Locate the passflt.dll file in the C:\winnt\system32 folder, and rename the file to **passflt.dll.old**.

11. Restart your machine for all changes to take effect.

Other Migration Scenarios

Because other deployment topologies are possible, your migration process may differ somewhat from the process described for a single-host deployment.

This section describes two alternative deployment scenarios and explains how to migrate each case. The sample deployment scenarios include:

- "Multimaster Replication Deployment"
- "Multi-Host Deployment with Windows NT" on page 233

Multimaster Replication Deployment

In a multimaster replication (MMR) deployment, two Directory Server instances are installed on different hosts. It is possible to run the hosts on different operating systems, but in this scenario, both hosts are running on the same operating system.

Table 7-2 illustrates how the Identity Synchronization for Windows components are distributed between the two hosts.

Table 7-2 Component Distribution in a Multimaster Replication Deployment

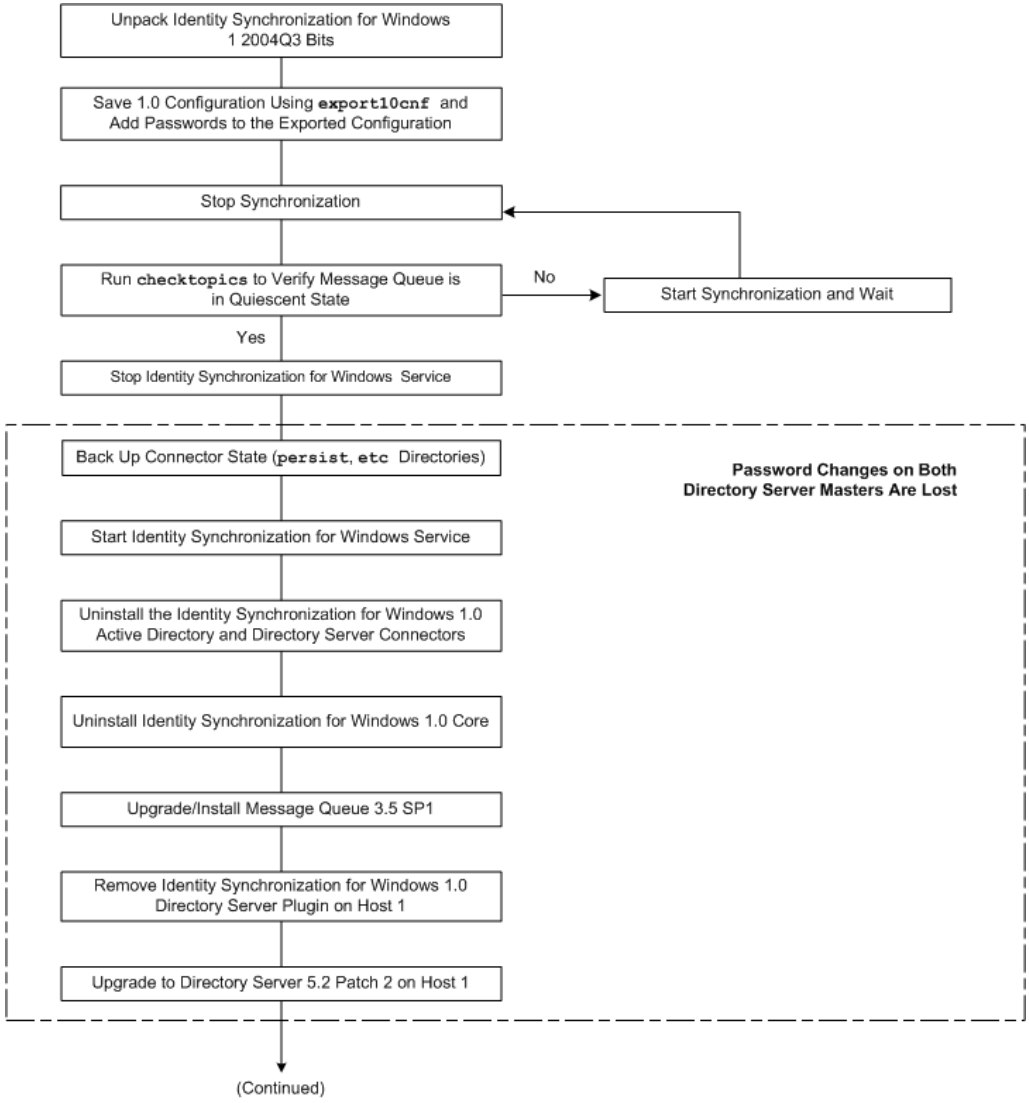
Host 1	Host 2
Directory Server (one instance) as the secondary master for synchronized users	Directory Server (one instance) as the preferred master for synchronized users
Core (Message Queue, Central Logger, System Manager, and Console)	Directory Server Plugin
Active Directory Connector	
Directory Server Connector	
Directory Server Plugin	

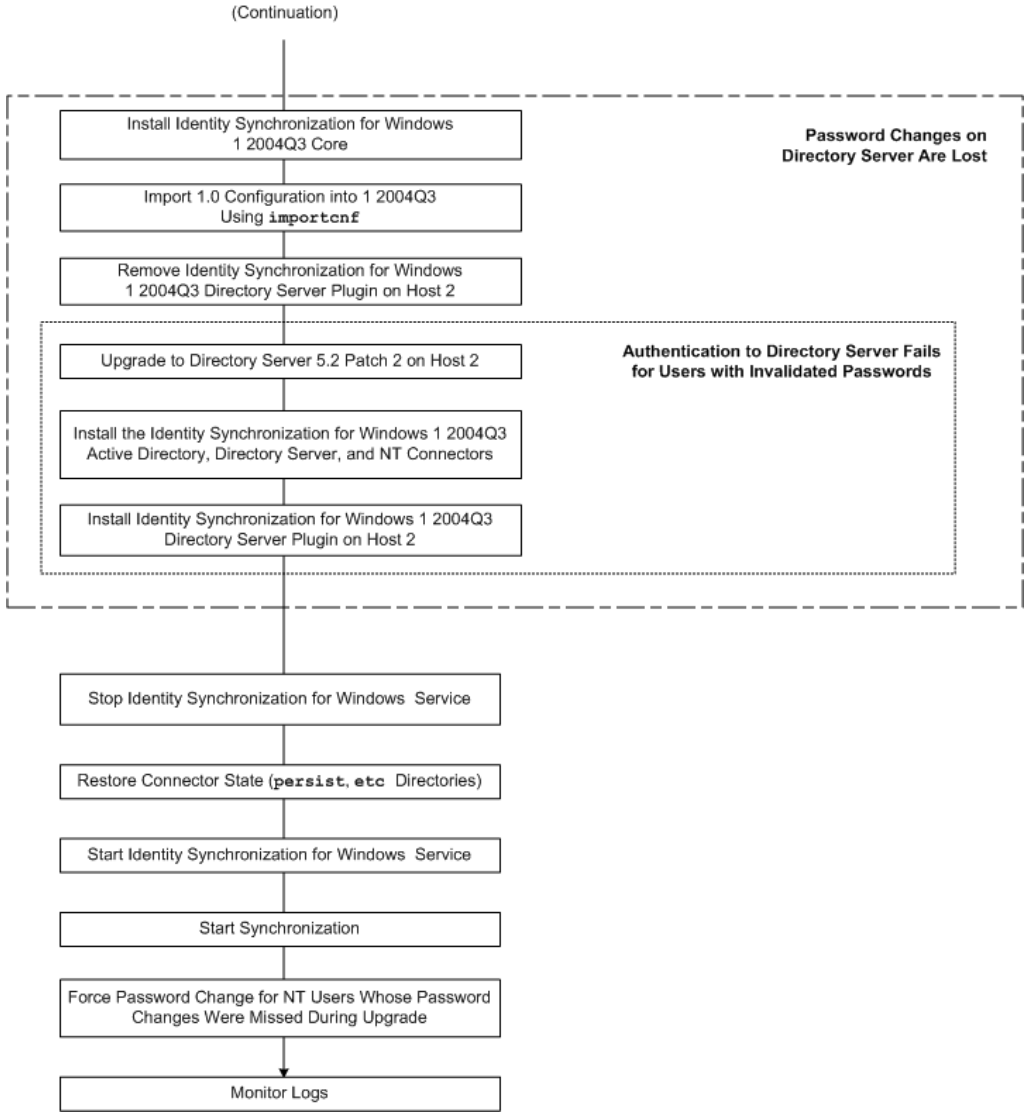
The migration process keeps on-demand password synchronization running continuously on the preferred master or on the secondary master.

NOTE If both hosts are running on a Solaris operating system, then a third host running Windows 2000 with Active Directory is required for synchronization purposes only. (No components would be installed on the third host.)

The following figure illustrates the process for migrating Identity Synchronization for Windows in a MMR deployment:

Figure 7-2 Migrating a Multimaster Replication Deployment




(Continuation)
Install Identity Synchronization for Windows 1 2004Q3 Core
Import 1.0 Configuration into 1 2004Q3 Using importcnf
Remove Identity Synchronization for Windows 1 2004Q3 Directory Server Plugin on Host 2
Upgrade to Directory Server 5.2 Patch 2 on Host 2
Install the Identity Synchronization for Windows 1 2004Q3 Active Directory, Directory Server, and NT Connectors
Install Identity Synchronization for Windows 1 2004Q3 Directory Server Plugin on Host 2
Password Changes on Directory Server Are Lost
Authentication to Directory Server Fails for Users with Invalidated Passwords
Stop Identity Synchronization for Windows Service
Restore Connector State (persist, etc Directories)
Start Identity Synchronization for Windows Service
Start Synchronization
Force Password Change for NT Users Whose Password Changes Were Missed During Upgrade
Monitor Logs

Multi-Host Deployment with Windows NT

Three hosts are used in this deployment scenario:

- A Windows NT system
- A host for Directory Server with the synchronized users and the Directory Server Connector
- A host for all other components

Table 7-3 illustrates how the Identity Synchronization for Windows components are distributed between the three hosts.

Table 7-3 Multi-Host Deployment

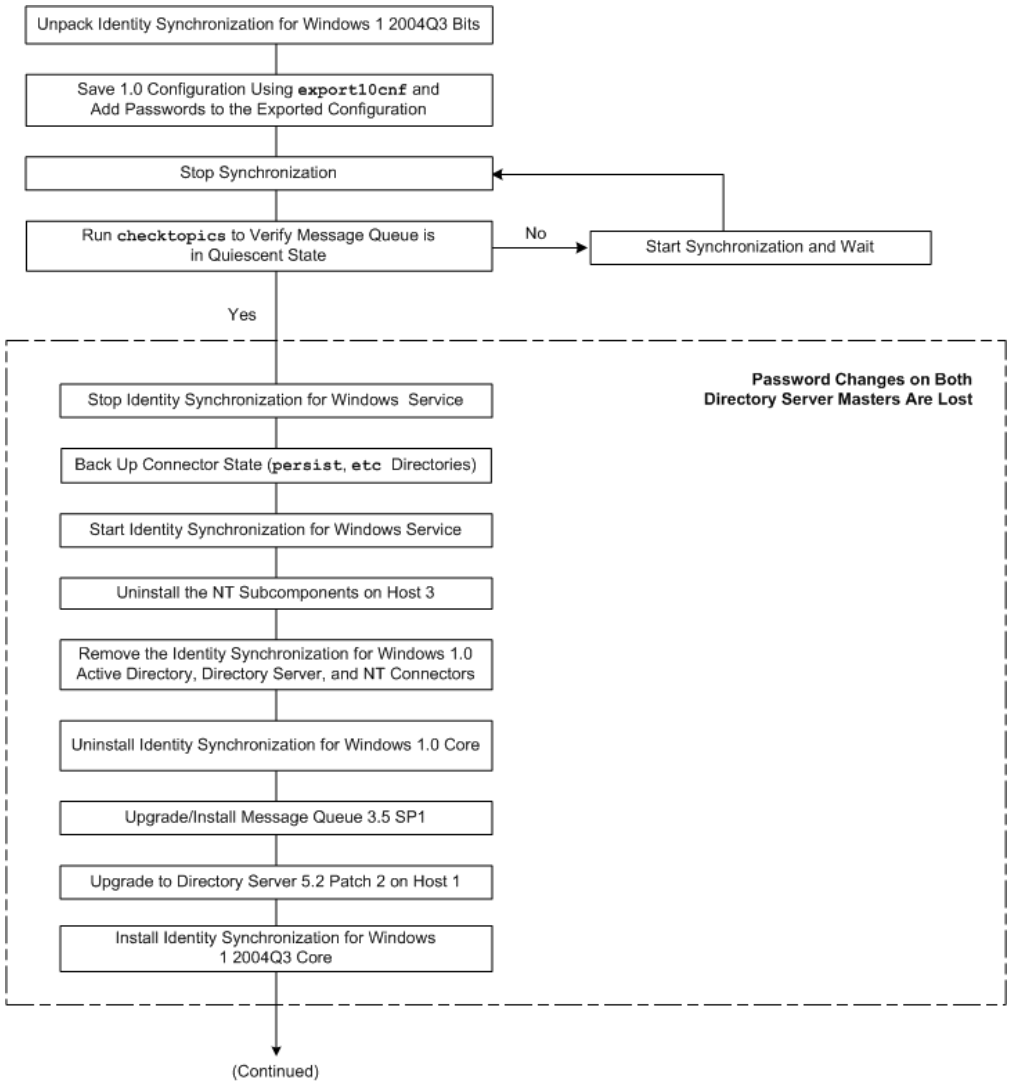
Host 1	Host 2	Host 3
Directory Server with configuration repository	Directory Server for synchronized users	Windows NT Connector
Core (Message Queue, Central Logger, System Manager, and Console)	Directory Server Connector	Windows NT Subcomponents (Password Filter DLL and Change Detector Service)
Active Directory Connector	Directory Server Plugin	

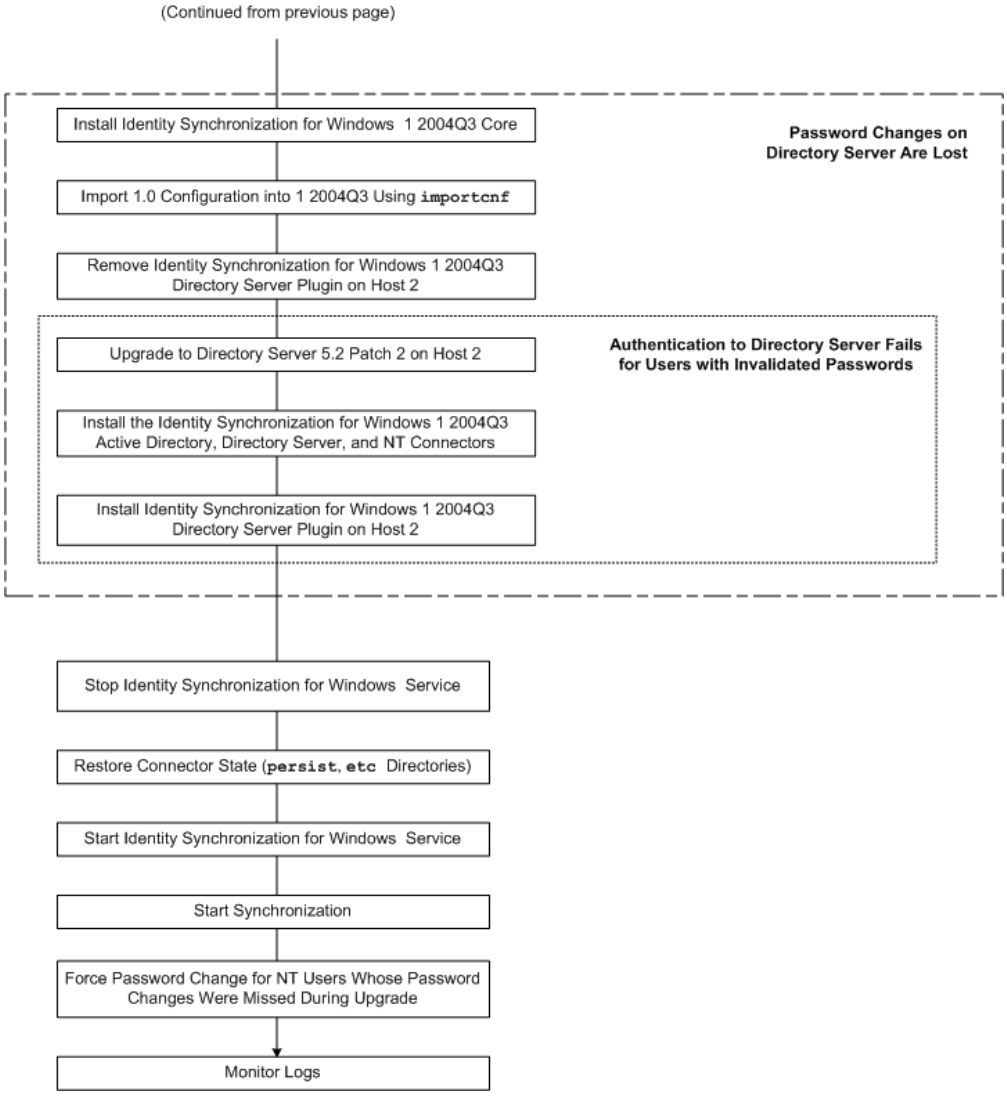
As in the previous scenario, Hosts 1 and 2 are running on the same operating system.

NOTE If both hosts are running a Solaris operating system, then a fourth host running Windows 2000 with Active Directory is required for synchronization purposes only. (No components would be installed on the fourth host.)

Figure 7-3 illustrates the process for migrating Identity Synchronization for Windows for a multi-host deployment:

Figure 7-3 Migrating a Multi-Host Deployment with Windows NT




(Continued from previous page)
Install Identity Synchronization for Windows 1 2004Q3 Core
Password Changes on Directory Server Are Lost
Import 1.0 Configuration into 1 2004Q3 Using importcnf
Remove Identity Synchronization for Windows 1 2004Q3 Directory Server Plugin on Host 2
Upgrade to Directory Server 5.2 Patch 2 on Host 2
Authentication to Directory Server Fails for Users with Invalidated Passwords
Install the Identity Synchronization for Windows 1 2004Q3 Active Directory, Directory Server, and NT Connectors
Install Identity Synchronization for Windows 1 2004Q3 Directory Server Plugin on Host 2
Stop Identity Synchronization for Windows Service
Restore Connector State (persist, etc Directories)
Start Identity Synchronization for Windows Service
Start Synchronization
Force Password Change for NT Users Whose Password Changes Were Missed During Upgrade
Monitor Logs

Checking the Logs

After migrating to version 1 2004Q3, check the central audit log for messages indicating a problem — especially for Directory Server users whose password changes may have been missed during the migration process, which would result in a message similar to the following:

```
[16/Apr/2004:14:23:41.029 -0500] WARNING    14  CNN101

ds-connector-host.example.com  "Unable to obtain password of user
cn=JohnSmith,ou=people,dc=example,dc=com, because the password was encoded
by a previous installation of Identity Synchronization for Windows
Directory Server Plugin. The password of this user cannot be synchronized at
this time. Update the password of this user again in the Directory Server."
```

You will not see this log message until after you start synchronization in Identity Synchronization for Windows 1 2004Q3, which is why checking the logs is the last step of the migration procedure.

Chapter 8

Removing the Software

This section contains procedures for removing Identity Synchronization for Windows 1 2004Q3 in the following sections:

Planning for Uninstallation

Before removing the software keep in mind the following points:

NOTE You must follow the instructions for uninstalling product components and subcomponents *explicitly*, and verify that you have uninstalled all components successfully.

- You must uninstall subcomponents and the Directory Server Plugin 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 Plugin before you uninstall Core.

  Uninstalling Core first will remove the Plugin 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 Plugin 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 NT 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 188).

- You must not uninstall Core unless all the connectors and subcomponents on all systems have been uninstalled.
- 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 run`Uninstall.sh` script (located in the installation directory, `/opt/SUNWisw`, by default) on Solaris operating systems. (You must run this script as root.)

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 Plugins
- Core

Your Windows NT system may contain the Windows NT Connector and subcomponents.

Use `runUninstaller.sh` (Solaris) or `uninstall.cmd` (Windows) to remove all connectors and subcomponents and then remove Core (if installed).

This section provides instructions for the following:

- Uninstalling the Directory Server Plugin
- Uninstalling Connectors
- Uninstalling Core

Uninstalling the Directory Server Plugin

NOTE

- The uninstaller removes Identity Synchronization for Windows Directory Server Plugins only. You cannot use this uninstaller to remove any other Directory Server Plugins.

  In this publication, *Directory Server Plugin* refers to the Identity Synchronization for Windows Directory Server Plugin (unless specifically noted otherwise).

- To run the uninstallation program in text-based mode (on Solaris only), type

  `./runUninstaller.sh -nodisplay`

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

Use the following steps to uninstall the Identity Synchronization for Windows Directory Server Plugin.

1. Start the uninstaller program (`runUninstaller.sh` on Solaris or `uninstall.cmd` on Windows).

   These uninstaller 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 Uninstall a Directory Server Plugin option.
6. Enter the Directory Server Host name, port, and your Administrator's credentials (name and password).
7. Click Next to perform further uninstallation related tasks.
8. When prompted restart the Directory Server where the Plugin was installed.
9. A summary window is displayed. Please follow the instructions presented in this window.
   - **On Solaris 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 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.

   Click Close to exit the program.
10. If the Directory Server Plugin is the *only* Identity Synchronization for Windows component installed on the target host, then you can delete the isw-*hostname* folder.
11. Repeat Step 1 through Step 9 for each Directory Server Plugin installed on a Windows 2000 server in your network.

Uninstalling Connectors

To uninstall connectors, use the following steps:

1. Start the uninstaller program (`runUninstaller.sh` on Solaris 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 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 Step 1 through Step 7 for all hosts where connectors are installed.

Uninstalling Core

NOTE You must uninstall the Directory Server Plugin before you uninstall Core.

Uninstalling Core before the Plugin removes the Plugin 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:
     I. Click `Start`, and then choose `Settings > Control Panel`.
     II. Double-click `Add/Remove Programs`.
     III. In the `Add/Remove Programs` window, select `Identity Synchronization for Windows`, then click `Remove`.

- On **Solaris or Windows** machines, execute `runUninstaller.sh` on Solaris 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 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.
   - **On Solaris 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 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 "Using resetconn" on page 326).

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 Systems

To uninstall the Console from a Solaris system, use the following steps:

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, use the following steps:

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`

Chapter 9

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:

Troubleshooting Checklist

NOTE Administrators: When you are debugging problems, adjust the logging level (as described in "Configuring Your Log Files" on page 277) to ensure the log reflects all events that may be causing 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 325), it displays a list of the remaining steps you have to perform to complete the installation and configuration process.

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. The Release Notes document many known issues. Is this problem explained there?

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 8, "Removing the Software" for more instructions about how to clean up previous installations.

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, 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 Plugin must be installed in each Directory Server replica.
10. Were post-installation procedures followed? The Directory Server must be restarted after the Directory Server Plugin 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 that existed in only one directory source, were these users created in the other directory source using the `idsync resync` command?

    NOTE You must run `idsync resync` whenever there are existing users. 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 Plugin 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 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

You can use one of the following methods to determine the connector ID:

- "Using the Central Logs"
- "Using idsync printstat"

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=airius,dc=com`
- **CNN100** is an Active Directory Connector that manages the `airius.com` domain

```
[2003/03/19 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=airius,dc=com
(ldap://host1.airius.com:389)];  CNN100 is the connector that manages
[airius.com (ldaps://host2.airius.com:636)];"
```

Using idsync printstat

The connector IDs and status are also available from the `idsync printstat` command (see “Using printstat” on page 325).

A sample output of this command follows:

```
Connector ID: CNN100
   Type:     Active Directory
   Manages:  airius.com (ldaps://host2.airius.com:636)
   State:    READY

Connector ID: CNN101
   Type:     Sun Java System Directory
   Manages:  dc=airius,dc=com (ldap://host1.airius.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.

```
[2003/03/19 10:20:16.889 -0600] INFO    13  SysMgr_100 host1  "Connector
[CNN101] is now in state "READY"."
```

Table 9-1 describes the different connector states.

Table 9-1 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 326) 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 261 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 256.

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 259.

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.
```

Troubleshooting Components

Use the information in this section to troubleshoot components. The information is organized as follows:

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 9-2 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 Plugin sends log records over the bus that are managed by the central logger for end-user viewing. However, the Plugin 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 Plugin's `logs` directory on the file system, which should look something like the following:

<serverroot>`/isw-`*<hostname>*`/logs/SUBC`*<id>*

Because the Plugin runs with the Directory Server process, there could potentially be a problem for the Plugin'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 Plugin 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 "Examining WatchList.properties" on page 258 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 `/var/sadm/install/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 this folder 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

Use the following checklist 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 Plugin 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 Plugin 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 Plugin 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
<no output>
```

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 '^]'.
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 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 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 free 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 the *Sun Java System 1 2004Q3 Identity Synchronization for Windows 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.

If the broker does run out of memory, follow these steps to recover:

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 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.[1] 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 "Starting and Stopping Services" on page 188.
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 the Chapter 11, "Configuring Security," which describes how to setup SSL between components in Identity Synchronization for Windows. This section contains:

- SSL Between Core Components
- SSL between Connectors and Directory Server or Active Directory
- SSL between the Directory Server Plugin and Active Directory

1. Even if all messages have been delivered, the broker might maintain up to 10000 message files to avoid the performance penalty of creating and deleting files.

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/2003:14:02:48.911 -0600] WARNING 14  CNN100 host1  "failed to open
connection to ldaps://host2.airius.com:636."
```

Open the Console and check the Specifying Advanced Security Options panel (see page 125).

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/2003:14:02:48.951 -0600] INFO    14  CNN100 host1  "failed to open
connection to ldaps://host2.airius.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.[1]

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`

1. Before running this command on Solaris, you must add the <*installation_root*>`/lib` directory to the `LD_LIBRARY_PATH` environment variable.

In this example, the certificate database contains no certificates:[1]

```
# /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
airius.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

1. The default certificate databases for the Sun Java System Directory Server and Windows NT Connectors include two certificates, `saint-cert100` and `saintRootCA`. These certificates are not used in this release.

```
# /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/2003:14:06:47.130 -0600] INFO    20  CNN100 host1  "failed to open
connection to ldaps://host2.airius.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 Plugin 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 11, "Configuring Security." If this certificate is not added, users will fail to bind to Directory Server with the error "DSA is unwilling to perform.", and the Plugin's log (for example, `isw-<`*hostname*`>/logs/SUBC100/pluginwps_log_0.txt`) will report:

```
[06/Nov/2003:15:56:16.310 -0600]  INFO   td=0x0376DD74 logCode=81
ADRepository.cpp:310    "unable to open connection to Active Directory
server at ldaps://host2.airius.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.

Chapter 10

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

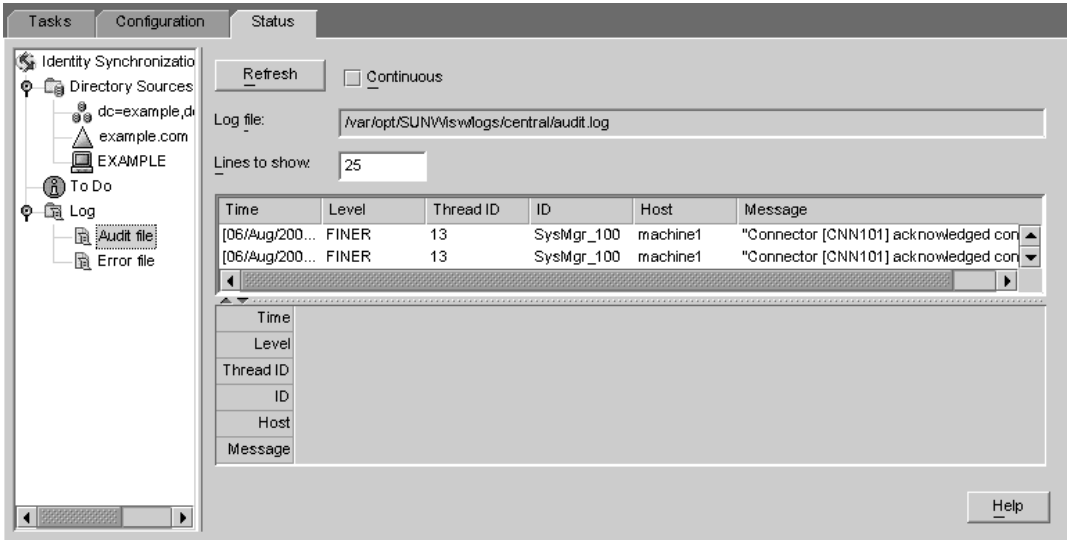
You can view various types of information from the Status tab of the Identity Synchronization for Windows Console.

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.)

Figure 10-1 The Status Tab



Log Types

This section describes the different kinds of logs that are available for Identity Synchronization for Windows:

- "Central Logs" on page 273
- "Local Component Logs" on page 274
- "Local Windows NT Subcomponent Logs" on page 274
- "Directory Server Plugin Logs" on page 275

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 Windows**: <*installation_root*>`/isw-`<*machine_name*>`/logs/central/`

The specific logs are described in Table 10-1.

Table 10-1 Identity Synchronization for Windows Log Types

Log Name	Description
`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 [airius.com (ldaps://
server1.airius.com:636)];  CNN101 is the connector that manages
[dc=airius,dc=com (ldap:// server2.airius.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 10-2 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 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 Plugin Logs

The Directory Server Plugin logs information through the Directory Server connector to the central log and through the Directory Server logging facility. Consequently, local Directory Server Plugin log messages will also be saved in the Directory Server error log.

Directory Server saves information into the error log from other Directory Server Plugins and components. To identify messages from the Identity Synchronization for Windows Directory Server Plugin, you can filter out lines containing the `isw` string.

By default, only minimal Plugin 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
- Plugins unable to establish connection to DS Connector at attila:1388,
will retry later
```

You can change the default verbosity level of the Directory Server error log from the Directory Server Management Console as follows:

1. Open the Directory Server Console.
2. Select the Configuration tab.
3. Click the Logs node in the navigation pane.
4. Select the Errors tab.
5. Click the Log Level button.
6. Enable the Plugins box. For extra verbosity, you can also enable Verbose Mode.
7. Click OK, and then click Save.

For more information about Directory Server logging, refer to the *Sun Java System Directory 5 2004Q2 Server Administrator's Guide* (`http://docs.sun.com/db/coll/DirectoryServer_04q2`).

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 10-3 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:

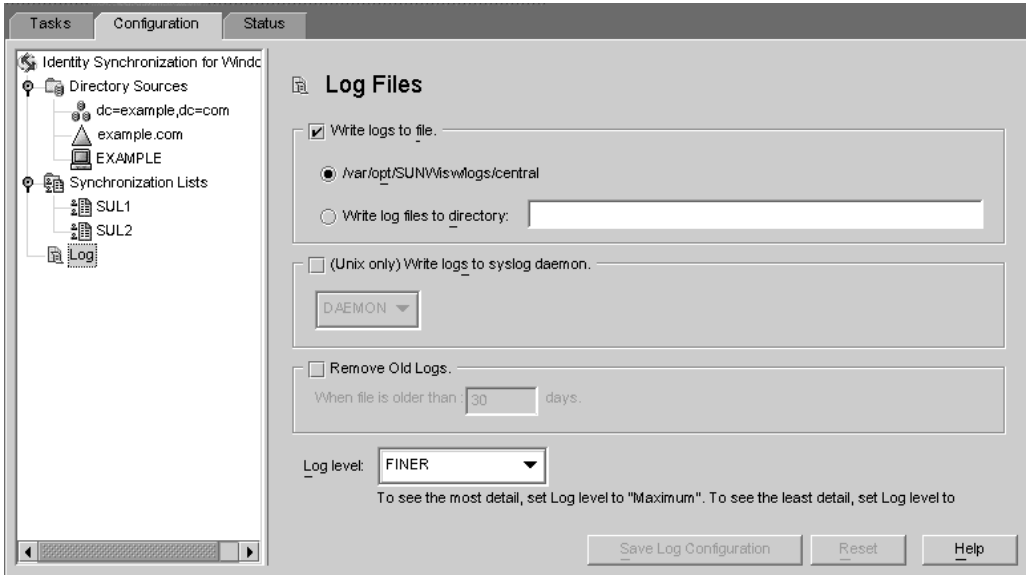
  “Resetting Central Logger configuration ...”
  “System manager is shutting down.”
  “Processing request (ID=<*ID_number*> from the console to stop synchronization.”

Configuring Your Log Files

You can use the Identity Synchronization for Windows Console to configure logging for your deployment, as follows:

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 (see Figure 10-2).

Figure 10-2 Configuring Log Files



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 "Log Levels" on page 276.)

5. Click the Save Log Configuration button to create log files based on the selected options.

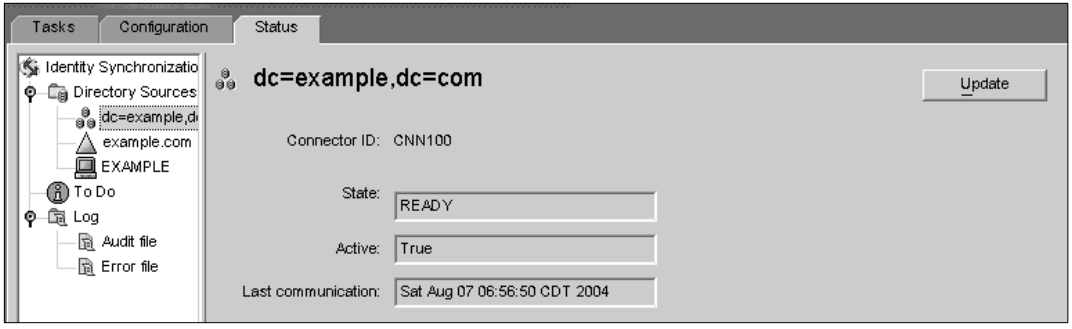
Viewing Directory Source Status

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 (for example, see Figure 10-3).

Figure 10-3 Directory Source Status



NOTE When viewing the Directory Source status you are essentially viewing the status of the connector associated with that Directory Source.

The following information is provided on the Status tab:

- **Update:** Click Update to refresh the information on this 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.

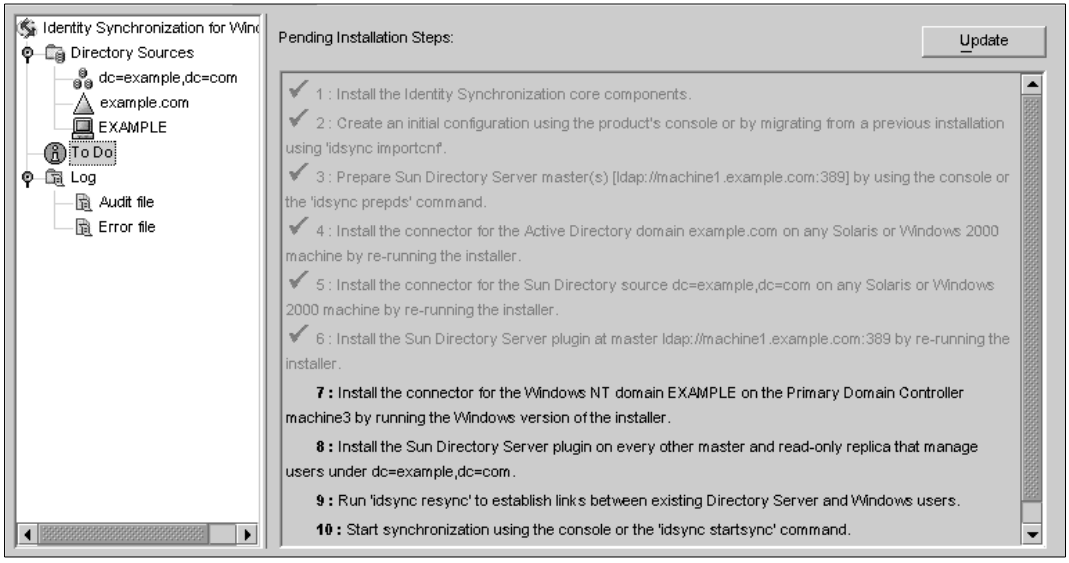
Viewing Installation and Configuration Status

To see which steps you must still complete in the Identity Synchronization for Windows installation and configuration process, use the following procedure:

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 Figure 10-3).

Figure 10-4 Viewing Your To Do List



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.

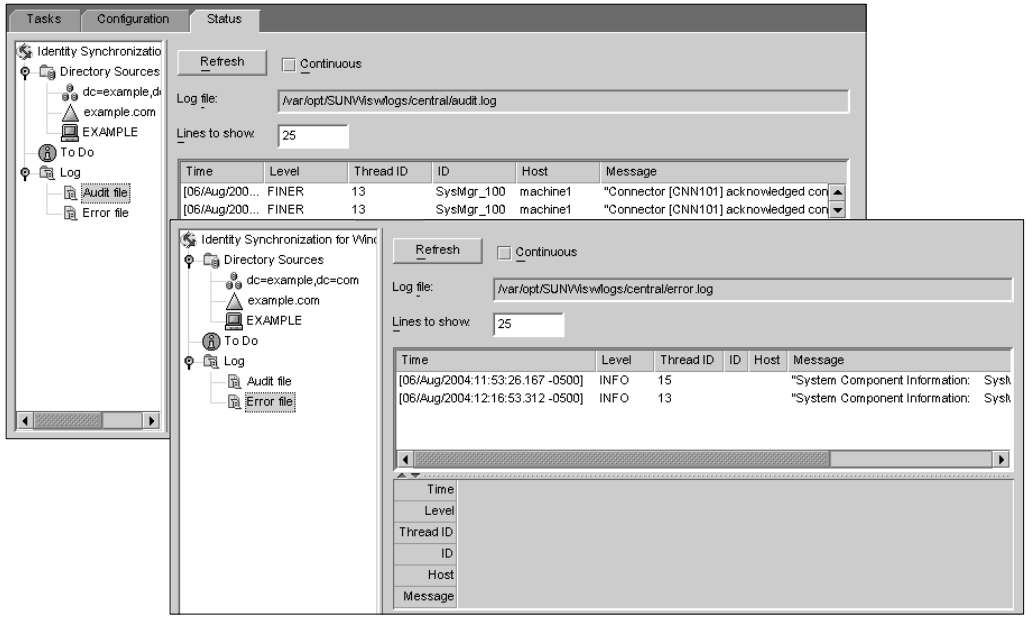
Viewing Your Audit and Error Logs

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 (Figure 10-5).

Figure 10-5 Viewing Your Logs



The following information is provided on the Status tab:

- **Refresh:** Loads the latest audit or error information.
- **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.

You can use the following procedure 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.

Chapter 11

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:

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 292.

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 page 125). 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 Plugin encrypts all user password changes with a 3DES key.

SSL and 3DES Keys Protection Summary

Table 11-1 summarizes how Identity Synchronization for Windows protects sensitive information that is sent over the network.

Table 11-1 Protecting Sensitive Information Using Network Security

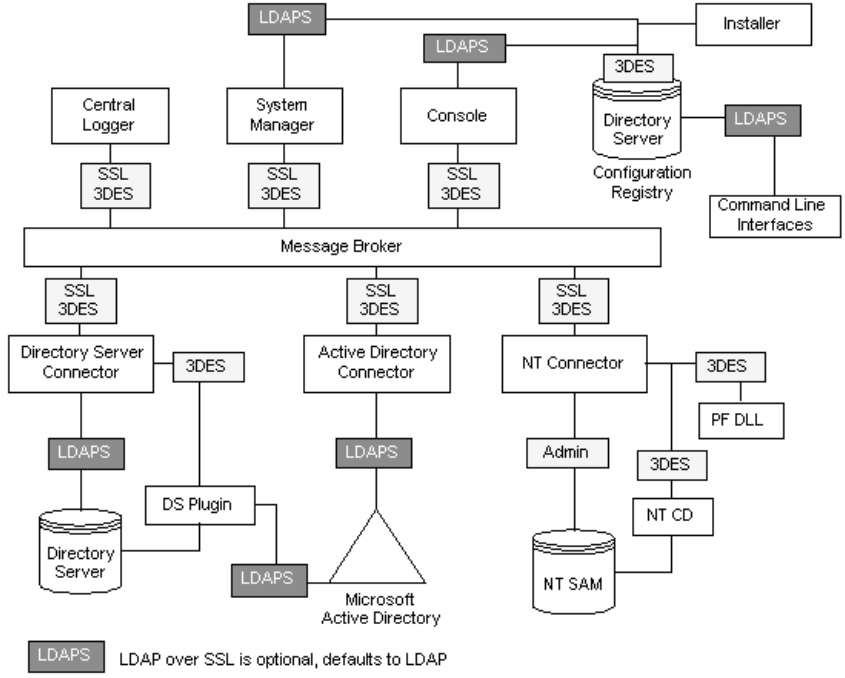
Use this Protection Method	Between the Following Information Types:
LDAP over SSL (optional)	• Directory Server Connector and Directory Server, Active Directory Connector and Active Directory • Directory Server Plugin and Active Directory • Command line interfaces and the product's configuration directory • 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

Table 11-1 Protecting Sensitive Information Using Network Security

Encrypted with 3DES keys (default)	• Directory Server Connector and Directory Server Plugin (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

Figure 11-1 contains an overview of the security features discussed in this section.

Figure 11-1 Identity Synchronization for Windows Security Overview



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 292 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

Table 11-2 summarize how Identity Synchronization for Windows protects sensitive information that is stored on disk.

Table 11-2 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" for recommendations to further protect the product's configuration directory.
Directory Server Retro Changelog	The Directory Server Plugin captures password changes and encrypts them before writing them to the Directory Server Retro Changelog.	The Directory Server Plugin 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.
Directory Server Plugin Boot Configuration	The Plugin'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.

Table 11-2 Persistent Storage Protection *(Continued)*

Persistent Storage	Confidential Information	Protection
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").

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

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 *Sun Java System Directory Server 5 2004Q2 Administrator's Guide,* Chapter 6: "Managing Access Control."

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.

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 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 Administrator's 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 *Sun Java System Directory Server 5 2004Q2 Administrator's Guide*, Chapter 6: "Managing Access Control."

Securing Replicated Configurations

Deployments connecting to Directory Servers using replication follow the same rules identified in Security Overview. 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 E, "Installation Notes for Replicated Environments."

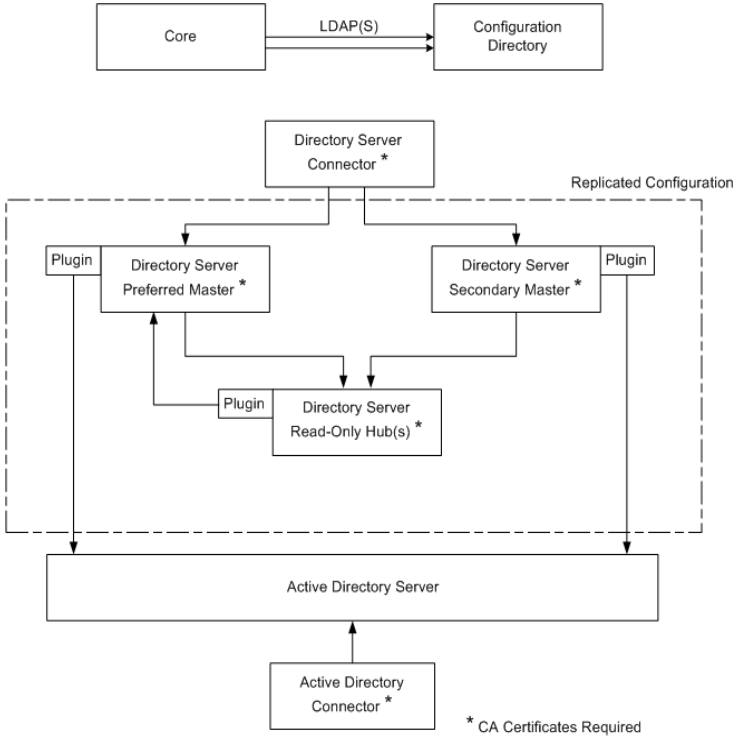
Table 11-3 lists the configuration components requiring CA certificates and identifies which certificates are required where.

Table 11-3 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

Figure 11-2 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 Plugin and there is only one Directory Server Connector, one Active Directory system, and one Active Directory Connector.

Figure 11-2 Replicated Configuration



NOTE 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 Plugin of type `other` that you install on a system with a Directory Server hub or read-only replica.

Directory Server Plugins have access to the same CA certificates as its associated Directory Server.

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 Plugins.

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

Table 11-4 describes the arguments you can use with the `idsync certinfo` subcommand:

Table 11-4 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`).

Table 11-4 certinfo Arguments *(Continued)*

Argument	Description
-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 *Directory Server 5 2004Q2 Administrator's Guide* for more information.

NOTE

- On Windows, use the `certutil` version bundled with Directory Server 5 2004Q2 (or later).

  Do not use the `certutil` shipped with Directory Server versions prior to 5 2004Q2. Earlier versions of `certutil` are incompatible with Identity Synchronization for Windows.

- On Solaris, `certutil` is installed in `/usr/sfw/bin` by default.

1. Create a new key certificate database for Directory Server by entering:

```
C:\Program Files\Sun\MPS\shared\bin\certutil.exe -d . -P slapd-hostname-

In order to finish creating your database, you
must enter a password which will be used to
encrypt this key and any future keys.
The password must be at least 8 characters long,
and must contain at least one non-alphabetic character.
Enter new password:
Re-enter password:
```

NOTE These examples are run in the `alias` directory immediately below the server root. Otherwise, Directory Server will not be able to find the certificate database.

2. Generate a self-signed certificate, which will be the server certificate used by Directory Server. Be sure you choose the subject DN according to the hostname of the server where Directory Server is running.

NOTE By default, a self-signed certificate is valid for three months. If you want to increase or decrease this time period, use the -v <*months-valid*> option. For example, to increase the time period to 24 months, enter **-v 21** or to decrease the time period to one month, enter **-v -2**.

```
C:\Program Files\Sun\MPS\shared\bin\certutil.exe -d . -P slapd-hostname-
-S -n server-cert -s "cn=hostname.example.com,c=us" -x -t CTu,,
A random seed must be generated that will be used in the
creation of your key.  One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.
To begin, type keys on the keyboard until this progress meter
is full.  DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!
Continue typing until the progress meter is full:
|************************************************************|
Finished.  Press enter to continue:
Enter Password or Pin for "NSS Certificate DB":
Generating key.  This may take a few moments...
```

3. Display the certificates for checking purposes.

```
C:\Program Files\Sun\MPS\shared\bin\certutil.exe -L -d . -P
slapd-hostname-
Certificate Name           Trust Attributes
server-cert                CTu,,
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
```

4. Create a `PIN` file, so that the certificate database password does not have to be entered each time you restart Directory Server.

   `C:\Program Files\Sun\MPS\alias > echo Internal (Software) Token:<`*`secret12`*`> slapd-`*`hostname`*`-pin.txt`

5. Enable SSL in the Directory Server as follows:

   a. Open the Console.

   b. Select the Configuration tab.

   c. Select the Encryption tab (on the right pane).

   d. Select Enable SSL for this server.

   e. Select Use this cipher family: RSA.

   f. Click Save and click OK twice.

   g. Select the Network tab.

   h. Update the Secure Port field. If running on the same machine as Active Directory, the port must be changed from 636 to an unused port or Directory Server will not start.

   i. Click Save, then yes, then OK.

   j. Select the Tasks tab (on the top).

   k. Click Restart Directory Server, then click yes.

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:

`C:\Program Files\Sun\MPS\shared\bin\certutil.exe -L -d . -P slapd-`*`hostname`*`- -n server-cert -a > C:\s-cert.txt`

These examples are run in the `alias` directory immediately below the server root. Otherwise, Directory Server will not find the certificate database.

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 302
- "Adding Active Directory Certificates to the Connector's Certificate Database" on page 305

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 Directory Certificate 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 the `cacert.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+Tu14NGoQnxhmNHLjANBgkqhkiG9w0BAQUFA
DCBjjEeMBwGCSqGSIb3DQEJARYPYmVydG9sZEBzdW4uY29tMQswCQYDVQQGEwJVUzELMAkG
A1UECBMCVFgxDzANBgNVBAcTBkF1c3RpbjEZMBcGA1UEChMQU3VuIE1pY3Jvc3lzdGVtczE
QMA4GA1UECxMHaVBsYW5ldDEUMBIGA1UEAxMLUmVzdGF1cmFudHMwHhcNMDIwMTExMDA1ND
A5WhcNMTIwMTExMDA1OTQ2WjCBjjEeMBwGCSqGSIb3DQEJARYPYmVydG9sZEBzdW4uY29tM
QswCQYDVQQGEwJVUELMAkGA1UECBMCVFgxDzANBgNVBAcTBkF1c3RpbjEZMBcGA1UEChMQU
3VuIE1pY3Jvc3lzdGVtczEQMA4GA1UECxMHaVBsYW5ldDEUMBIGA1UEAxMLUmVzdGF1cmFu
dHMwXDANBgkqhkiG9w0BAQEFAANLADBIAkEAyekZa8gwwhw3rLK3eV/12St1DVUsg31LOu3
CnB8cMHQZXlgiUgtQ0hm2kpZ4nEhwCAHhFLD3iIhIP4BGWQFjcwIDAQABo4IBnjCCAZowEw
YJKwYBBAGCNxQCBAYeBABDAEEwCwYDVR0PBAQDAgFGMA8GA1UdEwEB/wQFMAMBAf8wHQYDV
R0OBBYEFJ5Bgt6Oypq7T8Oykw4LH6ws2d/IMIIBMgYDVR0fBIIBKTCCASUwgdOggdCggc2G
gcpsZGFwOi8vL0NOPVJlc3RhdXJhbnRzLENOPWRvd2l0Y2hlcixDTj1DRFAsQ049UHVibGl
jJTIwS2V5JTIwU2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZmlndXJhdGlvbixEQz1yZX
N0YXVyYW50cyxEQz1jZW50cmFsLERDPXN1bixEQz1jb20/Y2VydGlmaWNhdGVSZXZvY2F0a
W9uTGlzdD9iYXNlP29iamVjdGNsYXNzPWNSTERpc3RyaWJ1dGlvblBvaW50ME2gS6BJhkdo
dHRwOi8vZG93aXRjaGVyLnJlc3RhdXJhbnRzLmNlbnRyYWwuc3VuLmNvbS9DZXJ0RW5yb2x
sL1Jlc3RhdXJhbnRzLmNybDAQBgkrBgEEAYI3FQEEAwIBADANBgkqhkiG9w0BAQUFAANBAL
5R9R+ONDdVHWu/5Sd9Tn9dpxN8oegjS88ztv1HD6XSTDzGTuaaVebSZV3I+ghSInsgQbH0g
W4fGRwaI 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” for instructions.

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.

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:
   - "Using Window's certutil" on page 302
   - "Using LDAP" on page 303
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:

     ```
     C:\\Program Files\Sun\MPS\shared\bin\certutil.exe -A -d . -n
     ad-ca-cert -t C,, -i \cacert.bin
     ```

   - If the certificate was retrieved using LDAP, type:

     ```
     C:\\Program Files\Sun\MPS\shared\bin\certutil.exe -A -d . -n
     ad-ca-cert -t C,, -a -i \ad-cert.txt
     ```

4. Restart the Identity Synchronization for Windows service / daemon.

NOTE Because the Directory Server `certutil.exe` is installed automatically when you install Directory Server 5 2004Q2, 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 5 2004Q2 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

Follow these steps to add the Active Directory CA certificate to the Directory Server certificate database.

NOTE Make sure that you have enabled SSL in Directory Server.

1. Retrieve the Active Directory CA certificate using one of the following methods:
   - “Using Window’s certutil” on page 302
   - “Using LDAP” on page 303
2. Stop Directory Server.
3. On the machine where Directory Server is installed, import the Active Directory CA certificate as follows:
   - If the certificate was retrieves using `certutil`, type:

     ```
     C:\\Program Files\Sun\MPS\shared\bin\certutil.exe -A -d . -P
     slapd-hostname- -n ad-ca-cert -t C,, -i \cacert.bin
     ```

- If the certificate was retrieves using LDAP, type:

  ```
  C:\\Program Files\Sun\MPS\shared\bin\certutil.exe -A -d . -P
  slapd-hostname- -n ad-ca-cert -t C,, -a -i \ad-cert.txt
  ```

4. Start Directory Server.

Adding Directory Server Certificates to the Directory Server Connector

If you enable SSL communication between the Directory Server Plugin and Active Directory, then you must add the Active Directory CA Certificate to the certificate database of each Directory Server master. Use the following steps:

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:

   ```
   C:\Program Files\Sun\MPS\shared\bin\certutil.exe -A -d . -n ds-cert -t
   C,, -i C:\s-cert.txt
   ```

   NOTE If the certificate was obtained in ASCII form, add an "`-a`" argument to the `certutil` command line to indicate that it is in ASCII form rather than binary form.

4. Restart the Identity Synchronization for Windows service/daemon.

Part II

Appendixes

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

This section describes the arguments (options) that are common to most of the command line utilities. The information is organized into the following tables:

- Table A-1 Arguments Common to All 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.

- Table A-2 SSL-Related Arguments Common to All Subcommands: 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.
- Table A-3 Configuration Directory Arguments: Describes arguments related to the configuration directory. These arguments are common to two or more `idsync` subcommands and migration tools.

NOTE Arguments that are unique to a particular subcommand will be explained in the pertinent subcommand section.

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-no*>	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, 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:**
  - **a.** Open a terminal window and **cd** to the /opt/SUNWisw/bin directory.
  - **b.** Type the idsync command with one subcommand, as follows

    idsync <*subcommand*>
- **From Windows:**
  - **a.** Open a Command Window and **cd** to the <*install_path*>\isw-<*hostname*>\bin directory.
  - **b.** Type the idsync command with one subcommand, as follows

    idsync <*subcommand*>

Table A-4 lists all of the idsync utility subcommands and their purpose:

Table A-4 idsync Subcommands Quick Reference

Subcommand	Purpose
certinfo	Displays certificate information based on your configuration and SSL settings (see "Using certinfo" on page 317)
changepw	Changes the Identity Synchronization for Windows configuration password (see "Using changepw" on page 318)
importcnf	Imports an exported Identity Synchronization for Windows version 1.0 configuration XML document (see "Using importcnf" on page 320)
prepds	Prepares a Sun Java System Directory Server source for use by Identity Synchronization for Windows (see "Using prepds" on page 321)
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 325)
resetconn	Resets connector states in the configuration directory to *uninstalled* (see "Using resetconn" on page 326)
resync	Links and resynchronizes existing users and pre-populates directories as part of the installation process (see "Using resync" on page 327)
startsync	Starts synchronization (see "Using startsync" on page 330)
stopsync	Stops synchronization (see "Using stopsync" on page 331)

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 Plugin 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" on page 312.

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:

Table A-5 idsync changepw Arguments

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.

NOTE For detailed information about other changepw arguments, review "Common Arguments" on page 312.

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 Step 2 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

CAUTION Use `idsync importcnf` *only* when migrating from Identity Synchronization for Windows 1.0 or 1.0 SP1 to version 1 2004Q3.

After installing Core (Chapter 3, "Installing Core"), use the `idsync importcnf` subcommand to import your exported Identity Synchronization for Windows version 1.0 (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-6 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" on page 312.

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 321) and then you can install the Identity Synchronization for Windows 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.

NOTE 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 Plugin, 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.

NOTE 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 Plugin 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:

```
idsync prepds [-D <bind-DN>] -w <bind-password | -> [-h <preferred host>]
[-p <preferred-port>] [-s <database-suffix>] [-Z] [-P <cert-db-path>]
[-m <secmod-db-path>] [-j <secondary_host>] [-r <secondary-port>] [-E <admin DN of
secondary host>] [-u <password for secondary host | ->] [-x]
```

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.

Table A-7 describes the arguments that are unique to idsync prepds:

Table A-7 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).
-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.

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 accomplishes the following:

- On M1:
  - 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 Plugin 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>`

NOTE For detailed information about the `printstat` arguments, review “Common Arguments” on page 312.

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.

CAUTION 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"
```

Table A-7 describes the arguments that are unique to resetconn:

Table A-8 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 directoy source names.

For detailed information about the other resetconn arguments, review "Common Arguments" on page 312.

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

NOTE For more detailed information about linking and synchronizing users, see “Synchronizing Existing Users” on page 179.

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>
```

Table A-9 describes the arguments that are unique to resync:

Table A-9 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, “LinkUsers XML Document Sample”)
`-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.
`-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

Table A-9 idsync resync Usage *(Continued)*

Argument	Meaning
-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" on page 312.
- For more information about resynchronizing existing users, review "Synchronizing Existing Users" on page 179.

After running `resync`, check the `resync.log` file in the central `audit log`. If errors result, consult Chapter 9, "Troubleshooting."

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>
```

Table A-10 describes the arguments that are unique to startsync:

Table A-10 idsync startsync Arguments

Argument	Description
[-y]	Does not prompt for command confirmation.

NOTE For detailed information about the other startsync arguments, review “Common Arguments” on page 312.

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" on page 312.

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 1 2004Q3 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 Plugin 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 (connector, Change Detector DLL, 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
```

Table A-11 describes the arguments that are unique to `forcepwchg`:

Table A-11 forcepwchg Arguments

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.

NOTE For more information about using `forcepwchg`, see "Forcing Password Changes on Windows NT" on page 201.

Appendix B

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 336 (an example of a common and simple configuration)
- “Sample 2: linkusers.cfg” on page 337 (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 command
    line.  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 'idsync resync' 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' 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's and sn's 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 Services as Non-Root on Solaris

You must have root privileges to install and run Identity Synchronization for Windows services. However, after installing the product you can configure the software to run the program services as a non-root user.

NOTE If you are going 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 on Solaris, perform the following steps:

1. Use the UNIX `useradd` command to create a user account for Identity Synchronization for Windows (*optional step*).

   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 on Solaris, you must choose a non-privileged port for the Connector during installation.
   (For example, ports larger than 1024 are acceptable.)

   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
   ```

Appendix D

Defining and Configuring Synchronization User Lists

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

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 an 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.

Table D-1 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 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.

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:

1. Map the Directory Server `destinationindicator` attribute to the Active Directory `activedirectorydomainname` attribute and to the Windows NT `user_nt_domain_name` attribute.
2. 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:

1. Create `cn=Jane Test,cn=users,dc=east,dc=example,dc=com` in the Active Directory east.example.com domain.
2. 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`.
3. Modify the `cn=Jane Test,ou=people,dc=example,dc=com` entry in the Directory Server.
4. 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, "LinkUsers XML Document Sample" for more information.

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.

Appendix E

Installation Notes for Replicated Environments

Identity Synchronization for Windows 1 2004Q3 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 5 2004Q2 Administrator's 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 5 2004Q2 Deployment Guide* to plan your deployment and the *Sun Java System Directory Server 5 2004Q2 Administrator's 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 a secondary master server for any given Sun directory source.

Directory Server version 5 2004Q2 now supports four-way MMR (where you can change the replicated database at any of the four masters). When you install the Plugin on a third or fourth master, you must select the *Other* host type and enter Directory Server instance's parameters manually during Plugin 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, proceed in the following order:

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 5 2004Q2 Administration Guide.*

To configure Directory Servers involved in replication so that all replication operations occur over an SSL connection, complete the following steps:

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

The following procedure summarizes the steps for configuring Identity Synchronization for Windows in an MMR Environment — detailed instructions are provided in other sections of this publication.

1. From the Identity Synchronization for Windows Console, specify a preferred and secondary Directory Server master for the suffix to be synchronized. (Review “Creating a Sun Java System Directory Source” on page 108.)

   You do not have to provide information about other Directory Servers in your topology.
2. Prepare the preferred and secondary servers from the Console or using the `idsync prepds` command line utility. (Review “Preparing the Directory Server” on page 115 or “Using prepds” on page 321.)

   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 165.)
4. Install the Directory Server Plugin on the preferred master, the secondary master, and every other Directory Server instance that manages users in the replicated suffix. (Review “Installing Directory Server Plugins” on page 175.)

Glossary

The following terms are used in the Identity Synchronization for Windows product and throughout this documentation set.

accessor A connector layer that interfaces directly with a directory source over protocols such as LDAP. Identity Synchronization for Windows has separate accessor implementations for Directory Server, Active Directory, and Windows NT. The accessor is often referenced in log messages about an action.

acknowledgement A specialized message that acknowledges receipt of a message from another component. Identity Synchronization for Windows uses acknowledgements between connectors and Message Queue, and between the connector components (agent, controller, and accessor) to ensure all changes are synchronized reliably.

action An encapsulation of a single synchronization event. Identity Synchronization for Windows Connectors use actions to communicate user change events. Each action includes a type (such as `CREATE`, `MODIFY`, or `DELETE`) and enough attributes from the user entry to allow the destination connector to synchronize the change. All actions are processed atomically.

agent A connector component that interfaces with Message Queue and translates attributes between their Directory Server names and Windows names. The agent is often referenced in log messages about an action.

attribute Holds descriptive information about an entry. Attributes have a label and a value. Each attribute also follows a standard syntax for the type of information that can be stored as the attribute value.

attribute list A list of required and optional attributes for a given entry type or object class.

audit log A central log file that contains entries for day-to-day events, such as a user's password being synchronized. Administrators can use the Identity Synchronization for Windows Console to control how many entries and what level of detail will be displayed in this log.

Each connector produces an audit log of the users processed by that connector, and there is a centralized audit log containing an aggregation of the audit logs produced by all of the connectors in your deployment.

authentication Process of proving the identity of the client user to Directory Server. Users must provide a bind DN and the corresponding password to be granted access to the directory. Directory Server allows the user to perform functions or access files and directories based on the permissions granted to that user by the directory administrator.

authentication certificate A digital file, issued by a third party, that cannot be transferred or forged. Authentication certificates are sent from server to client (or from client to server) to verify and authenticate the other party.

Auxiliary objectclass An objectclass that augments the selected structural class, which provides additional attributes for synchronization. See Structural object class.

base DN Base distinguished name. A search operation is performed on the base DN, the DN of the entry, and all entries below it in the directory tree. For Active Directory and Directory Server, Synchronization User Lists are rooted at a specific base DN. All users under this base DN will be synchronized unless they are explicitly excluded by a filter.

base distinguished name See base DN.

bind DN Distinguished name used to authenticate to an LDAP directory (e.g. Active Directory or Directory Server) when performing an operation.

bind distinguished name See bind DN.

Broker See Sun Java System Message Queue Broker.

CA See certificate authority.

cascading replication In a cascading replication scenario; one server (often called the *hub supplier*) acts both as a consumer and a supplier for a particular replica. The server holds a read-only replica and maintains a change log. It receives updates from the supplier server that holds the master copy of the data, and in turn supplies those updates to the consumer.

central logger A Core component that manages all of the central logs, which are an aggregation of every connector's audit and error logs. Administrators can monitor the health of an entire Identity Synchronization for Windows installation by monitoring these logs. You can view the central logs directly or from the Identity Synchronization for Windows Console. By default, the central logs are available on the machine where Core was installed under the <*install-root*>/logs/central/ subdirectory.

certificate A collection of data that associates public keys with a network identity. This information enables the recipient of an electronic message to verify the authenticity of the message and the message sender. When you configure Identity Synchronization for Windows Connectors to use SSL communication, you must add certificates to the connector's certificate databases before trusted SSL communication can occur. See also certificate authority.

certificate authority A company or organization that sells and issues authentication certificates. You may purchase an authentication certificate from a certificate authority (also known as a *CA*) that you trust. A root certificate authority certificate is used to sign other certificates. When configuring an Identity Synchronization for Windows Connector to use SSL, you must add the appropriate root certificate authority certificate to the Connector's certificate database.

certificate database A secure repository for certificates, which includes three files: cert8.db, key3.db, and secmod.db. In Identity Synchronization for Windows, each connector has its own certificate database directory (for example, <*install-root*>/etc/CNN100). See also certificate.

character type Distinguishes alphabetic characters from numeric (or other) characters and the mapping of upper-case to lower-case letters.

CLI See command line interface

client See LDAP client.

command line interface A means of communication between a program and its user, based solely on textual input and output. Commands are input with the help of a keyboard or similar device, and are interpreted and executed by the program. The Identity Synchronization for Windows command line interface is named `idsync` and is available in the `bin/` directory where you installed Core.

configuration directory A special installation of Directory Server that serves as a repository for configuration and status information. Identity Synchronization for Windows stores all of its configuration within the configuration directory instance chosen during Core installation.

configuration password A password chosen during Core installation that protects all sensitive Identity Synchronization for Windows information stored in the configuration directory. The configuration password must be provided when using the installer, the console, or the command line interface.

configuration registry Another term used by Identity Synchronization for Windows to refer to the configuration directory.

connector A Java process that manages Identity Synchronization for Windows' interaction with a single data source (such as a Directory Server, an Active Directory domain, or a Windows NT domain). A connector is responsible for detecting user changes in the data source and publishing these changes to remote connectors over Message Queue, and for subscribing to user change topics and applying updates from these topics to the data source.

console A Graphical User Interface used to configure and monitor server applications. The Sun Java System Directory Server and Identity Synchronization for Windows have separate consoles.

controller A connector component that interfaces with the agent and accessor components. The controller performs key synchronization-related tasks such as determining a user's membership in a Synchronization User List, searching for and linking equivalent user entries, and detecting changes to users by comparing current user entries with the previous versions stored in the object cache. The controller is often referenced in log messages about an action.

Core The first Identity Synchronization for Windows component that is installed. The Core includes the initial configuration stored in the configuration directory, the System Manager, the Central Logger, the console, and the command line interface.

creation attributes Attributes that are synchronized only when an object is created. All significant attributes are automatically synchronized when an object is created. You can configure default values for creation attributes that might not have a corresponding attribute value in the remote directory.

daemon A background process on a UNIX machine that is responsible for a particular system task. Daemon processes do not need human intervention to continue functioning. Connectors, the system manager, and the central logger run as daemon processes that are launched and monitored by the Identity Synchronization for Windows Watchdog.

directory information tree The logical representation of the information stored in the directory that mirrors the tree model used by most file systems, where the tree's root appears at the top of the hierarchy.

Directory Manager The privileged directory server administrator, comparable to the root user in UNIX. Identity Synchronization for Windows requires Directory Manager credentials to perform certain configuration operations, but the connector does not require Directory Manager credentials for synchronization.

directory source A Sun Java System Directory Server, Windows Active Directory Domain, or Windows NT Domain. Directory sources contain users to be synchronized.

distinguished name String representation of an entry's name and location in an LDAP directory.

DIT See directory information tree.

DM See Directory Manager.

domain (1) (n.) The last part of a fully qualified domain name that identifies the company or organization that owns the domain name (for example, example.com, host.example.com).

(2) (n.) Resources under control of a single computer system.

domain controller A Windows server that stores user account information, authenticates users, and enforces security policy for a Windows domain. Identity Synchronization for Windows Connectors communicate directly with domain controllers to detect changes to user accounts and to synchronize changes made in Directory Server user entries.

DNS Domain Name System. System used by machines on a network to associate standard IP addresses (such as 198.93.93.10) with hostnames (such as `www.example.com`). Machines normally get the IP address for a hostname from a DNS server or look up the address in tables maintained on their systems.

file extension Portion of a filename following the period or dot (.) that typically defines the file type (for example, .GIF and .HTML). For example, in a file named `index.html` the file extension is *`html`*.

file type The format of a given file. For example, graphics files are often saved in `GIF` format, while a text file is usually saved as `ASCII` text format. File types are usually identified by the file extension (for example, .GIF or .HTML).

FSMO Role Flexible Single-Master Operation role. Mechanism used by Active Directory to prevent update conflicts in multimaster deployments. Some objects are updated in a single-master mode even if the deployment is multimaster, which is very similar to the old concept of a Primary Domain Controller (PDC) in Windows NT domains. There are five FSMO Roles in an Active Directory deployment, but only the PDC-emulator role affects Identity Synchronization for Windows. Because password updates are replicated immediately only to the Active Directory domain control with the PDC emulator role, Identity Synchronization for Windows use this domain controller for synchronization. Otherwise, synchronization with the Sun Java System Directory Server might be delayed for several minutes.

global catalog A Windows repository that stores Active Directory directory topology and schema information for Active Directory directories.

hostname A name for a machine in the form machine.domain.com, which is translated into an IP address. For example, `www.example.com` is the machine *www* in the subdomain *example*, and domain *com*.

Identity Synchronization for Windows Console A Graphical User Interface used to configure and monitor Identity Synchronization for Windows.

inbound Within the connector, the direction of actions that flow from a directory source toward Message Queue. Changes detected by the connector flow inbound into the system. Log messages about an action often refer to events that occur on the inbound side of the connector.

IP address Internet Protocol address. A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 192.168.2.1).

ISO International Standards Organization.

Java Message Service A messaging standard API that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous.

JMS See Java Message Service.

LDAP Lightweight Directory Access Protocol. Directory service protocol designed to run over TCP/IP and across multiple platforms. Identity Synchronization for Windows uses LDAP to communicate with Active Directory domain controllers and Sun Java System Directory Servers.

LDAP client Software used to request and view LDAP entries from an LDAP Directory Server. Identity Synchronization for Windows Connectors act as LDAP clients when connecting to LDAP servers.

LDAP URL Provides the means of locating directory servers using DNS and then completing the query via LDAP. A sample LDAP URL is `ldap://ldap.example.com`

Lightweight Directory Access Protocol See LDAP.

locale Identifies the collation order, character type, monetary format, and time / date format used to present data for users of a specific region, culture, and/or custom. This includes information on how data of a given language is interpreted, stored, or collated. The locale also indicates which code page should be used to represent a given language.

main object class See Structural object class.

Message Queue See Sun Java System Message Queue

MMR See multimaster replication.

MQ See Sun Java System Message Queue.

multimaster replication A directory server replication model in which entries can be written and updated on any of several master replica copies without requiring communication with other master replicas before the write or update is performed. Modifications made on one server are automatically replicated to the other servers. Identity Synchronization for Windows can be installed in a

deployment with multiple directory server masters. However, when synchronizing changes to Windows, the preferred directory server must be available, and when synchronizing changes from Windows, the preferred or secondary directory server must be available.

naming context (also knows as root suffix) A specific suffix of a directory information tree (DIT) that is identified by its distinguished name (DN), e.g. dc=example,dc=com. In Identity Synchronization for Windows, a directory source for Sun Java System Directory Server is defined by the suffix containing the data to be synchronized.

object cache An in-process database used by the Windows Connectors to detect changes to user entries. The object cache stores a hashed summary of each user entry, which enables Windows Connectors to determine which specific attributes in the user entry have changed.

object class A template specifying the kind of object that the entry describes and the set of valid and mandatory attributes that entry contains. For example, Directory Server specifies an `inetorgperson` object class which has attributes such as `cn` and `userpassword`.on-demand password synchronization: a mechanism whereby a user's password in Directory Server is not updated until the user attempts to authenticate to Directory Server. The user's password is synchronized only if the provided password matches what is stored in Active Directory. This simplifies password synchronization in Active Directory environments.

outbound Within the connector, the direction of actions that flow from Message Queue toward the directory source. Changes applied by a connector flow outbound into the synchronized directory source. Log messages about an action often refer to events that occur on the outbound side of the connector.

password file A file on UNIX machines that stores UNIX user login names, passwords, and user ID numbers. It is also known as `/etc/passwd`, because of its location.

password policy A set of rules that govern how passwords are used in a given directory.

permission In the context of access control, the permission states whether access to the directory information is granted or denied, and the level of access that is granted or denied.

plug-in An accessory program that can be loaded and then used as part of the overall system.

For example, Identity Synchronization for Windows uses the Directory Server Plugin to enhance the Directory Server Connector change-detection features and to provide bidirectional support for password synchronization between Active Directory and Directory Server.

preferred directory server A directory server master instance used by Identity Synchronization for Windows to detect and apply changes to user entries. While this server is available, Identity Synchronization for Windows will not communicate with any other directory server masters.

protocol A set of rules that describes how devices on a network exchange information.

RCL Se retro changelog.

resync interval How often a connector checks a directory source for changes. This periodic check is efficient and only requires reading entries of users that have changed since the last check. The console expresses this value in milliseconds and provides 1000 (1 second) as a default.

retro changelog A Directory Server database (cn=changelog) that stores a record of all changes made to Directory Server. Identity Synchronization for Windows uses the retro changelog to detect changes made to Directory Server. In an MMR environment, the retro changelog must be enabled on the Preferred Directory Server.

root The most privileged user available on UNIX machines (also called superuser). The root user has complete access privileges to all files on the machine. On Solaris systems, Identity Synchronization for Windows must be installed as root.

root suffix The parent of one or more LDAP sub-suffixes. A directory tree can contain more than one root suffix.

schema Definitions describing what types of information can be stored as entries in the directory. When information that does not match the schema is stored in the directory, clients attempting to access the directory may be unable to display the proper results.

schema checking Ensures that entries added or modified in the directory conform to the defined schema. Schema checking is on by default and users will receive an error if they try to save an entry that does not conform to the schema.

secondary directory server A directory server master instance in an MMR environment that Identity Synchronization for Windows can use when the preferred directory server is not available. While the preferred directory server is unavailable, Identity Synchronization for Windows can synchronize changes made in Active Directory or Windows NT to the secondary directory server, but changes made at the secondary server or any other directory server master will not be synchronized until the preferred directory server is available.

Secure Sockets Layer See SSL.

Server Console Java-based application that allows you to perform administrative management of your Directory Server from a GUI.

server root A directory on the server machine dedicated to holding the server program configuration, maintenance, and information files.

service A background process on a Windows machine that is responsible for a particular system task. Service processes do not need human intervention to continue functioning. On Windows, connectors, the system manager, and the central logger run as processes that are launched and monitored by the Identity Synchronization for Windows Watchdog service.

significant attributes Attributes that are synchronized when an entry is created or modified.

SSL Secure Sockets Layer. A software library used for establishing a secure connection between two parties (client and server). Used to implement HTTPS, the secure version of HTTP, and LDAPS the secure version of LFAP.

Structural object class The primary object class of an entry that defines the set of valid and mandatory attributes on the user entries that Identity Synchronization for Windows synchronizes. For example, the default Active Directory object class is `user`, and the default Directory Server object class is `inetorgperson`. See Auxiliary objectclass.

subcomponent A lightweight process or library that runs separate from a connector. A subcomponent runs close to the directory source that a connector manages, and enables functionality in the connector that cannot be achieved in a remote machine or separate process. The subcomponent communicates with connector over a custom encryption channel to receive configuration information, report change events, and log to the central logger. Identity Synchronization for Windows includes three subcomponents: the Directory Server Plugin, the Windows NT Password Filter DLL, and the Windows NT Change Detector.

suffix The name of the entry at the top of the directory tree, below which data is stored. Multiple suffixes are possible within the same directory. Each database has only one suffix.

SUL See Synchronization User List.

Sun Java System Message Queue An enterprise messaging system that implements the Java Message Service (JMS) open standard. The basic architecture of Message Queue consists of publishers and subscribers that exchange messages by way of a common service. The Sun Java System Message Queue is administered by a dedicated message broker, which is responsible for controlling access to Message Queue, maintaining information about active publishers and subscribers, and ensuring that messages are delivered. Identity Synchronization for Windows uses Message Queue to securely synchronize user change events, distribute configuration information, and monitor the health of remote components.

Sun Java System Message Queue Broker A standalone Java server that provides clients access to the Sun Java System Message Queue. On Solaris, the Broker is controlled via the `/etc/init.d/imq daemon` script, and on Windows, it is controlled via the "iMQ Broker" service. Identity Synchronization for Windows configures and starts the broker during Core installation.

superuser See root.

synchronization host Servers that store synchronized data according to the rules defined in the Synchronization User Lists (SULs).

Synchronization User List Defines users in the Sun and Windows directories to be synchronized. A Synchronization User List can restrict the scope of users to be synchronized based on an LDAP base DN or filter.

synchronized attributes See significant attributes.

System Manager A stand-alone Java process that is started by the Watchdog daemon/service where Core is installed. The system manager distributes configuration information to the connectors and central logger, monitors the health of the system, and coordinates `idsync resync` operations.

topology The way a directory tree is divided among physical servers and how these servers link with one another.

uid A unique number associated with each user on a UNIX system.

URL Uniform Resource Locator. The addressing system used by the server and the client to request documents. It is often called a location. The format of a URL is `[protocol]://[machine:port]/[document]`. The port number is necessary only on selected servers, and it is often assigned by the server, freeing the user of having to place it in the URL.

Watchdog A stand-alone Java process that is installed on every machine where Core or a connector is installed. The Watchdog starts all Identity Synchronization for Windows Java processes including the System Manager, the Central Logger, and Connectors. If any of these components fail, the Watchdog restarts them. On Solaris, the Watchdog is controlled via the `/etc/init.d/isw` daemon script, and on Windows, it is controlled via the "Sun Java™ System Identity Synchronization for Windows" service.

Index

NUMERICS

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X