



Sun Java System Identity Synchronization for Windows 6.0 Installation and Configuration Guide



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

Part No: 821-0422
November 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more U.S. patents or pending patent applications in the U.S. and in other countries.

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, the Solaris logo, the Java Coffee Cup logo, docs.sun.com, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. or its subsidiaries in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and SunTM Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs brevets américains ou des applications de brevet en attente aux Etats-Unis et dans d'autres pays.

Cette distribution peut comprendre des composants développés par des tierces personnes.

Certains composants de ce produit peuvent être dérivées du logiciel Berkeley BSD, licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays; elle est licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, le logo Solaris, le logo Java Coffee Cup, docs.sun.com, Java et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc., ou ses filiales, aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la législation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

- Preface11**

- Part I Installing Identity Synchronization for Windows 21**

- 1 Understanding the Product23**
 - Product Features 24
 - System Components 25
 - Watchdog Process 26
 - Core 27
 - Connectors 29
 - Connector Subcomponents 30
 - Message Queue 31
 - System Components Distribution 31
 - Core 32
 - Directory Server Connector and Plug-in 32
 - Active Directory Connector 32
 - Windows NT Connector and Subcomponents 33
 - How Identity Synchronization for Windows Detects Changes in Directory Sources 34
 - How Directory Server Connectors Detect Changes 35
 - How Active Directory Connectors Detect Changes 35
 - How Windows NT Connectors Detect Changes 36
 - Propagating Password Updates 37
 - Reliable Synchronization 39
 - Deployment Example: A Two-Machine Configuration 40
 - Physical Deployment 42
 - Component Distribution 42

2	Preparing for Installation	45
	Installation Overview	45
	Installing Core	47
	Configuring the Product	47
	Preparing the Directory Server	48
	Installing Connectors and Configuring Directory Server Plug-In	48
	Synchronizing Existing Users	49
	Configuration Overview	49
	Directories	50
	Synchronization Settings	50
	Object Classes	50
	Attributes and Attribute Mapping	51
	Synchronization User Lists	52
	Synchronizing Passwords With Active Directory	53
	Enforcing Password Policies	54
	Configuring Windows for SSL Operation	59
	Installation and Configuration Decisions	60
	Core Installation	60
	Core Configuration	60
	Connector Installation and Configuring the Directory Server Plug-In	61
	Using the Command-Line Utilities	62
	Installation Checklists	63
3	Installing Core	67
	Before You Begin	67
	Starting the Installation Program	68
	On Solaris SPARC	68
	On Solaris x86	68
	On Windows	69
	On Red Hat Linux	69
	Installing Core	70
	▼ To Install Identity Synchronization for Windows Core Components Using the Installation Wizard	70

4	Configuring Core Resources	79
	Configuration Overview	79
	Opening the Identity Synchronization for Windows Console	80
	▼ To Open Identity Synchronization for Windows Console	81
	Creating Directory Sources	84
	▼ To Create Directory Sources	84
	Creating a Sun Java System Directory Source	85
	Preparing Sun Directory Source	91
	Creating an Active Directory Source	95
	Creating a Windows NT SAM Directory Source	102
	Selecting and Mapping User Attributes	105
	Selecting and Mapping Attributes	105
	Creating Parameterized Default Attribute Values	107
	Changing the Schema Source	108
	Propagating User Attributes Between Systems	110
	Specifying How Object Creations Flow	111
	Specifying How Object Modifications Flow	116
	Specifying Configuration Settings for Group Synchronization	124
	Configuring and Synchronizing Account Lockout and Unlockout	125
	Specifying How Deletions Flow	128
	Creating Synchronization User Lists	129
	▼ To Identify and Link User Types Between Servers	129
	Saving a Configuration	133
	▼ To Save your Current Configuration from the Console Panels	133
5	Installing Connectors	137
	Before You Begin	137
	Running the Installation Program	138
	▼ To Restart and Run the Installation Program	138
	Installing Connectors	140
	Installing the Directory Server Connector	140
	Installing an Active Directory Connector	146
	Installing the Windows NT Connector	148

6	Synchronizing Existing Users and User Groups	151
	Post-Installation Steps Based on Existing User and Group Populations	152
	Using idsync resync	152
	Resynchronizing Users or Groups	152
	Linking Users	153
	idsync resync Options	154
	Checking Results in the Central Log	157
	Starting and Stopping Synchronization	158
	▼ To Start or Stop Synchronization	158
	Resynchronized Users/Groups	159
	Starting and Stopping Services	159
7	Removing the Software	161
	Planning for Uninstallation	161
	Uninstalling the Software	162
	Uninstalling Connectors	162
	▼ To Uninstall Core	164
	Uninstalling the Console Manually	166
	From Solaris or Linux Systems	167
	From Windows Systems	167
8	Configuring Security	169
	Security Overview	169
	Specifying a Configuration Password	170
	Using SSL	170
	Requiring Trusted SSL Certificates	171
	Generated 3DES Keys	171
	SSL and 3DES Keys Protection Summary	171
	Message Queue Access Controls	173
	Directory Credentials	174
	Persistent Storage Protection Summary	174
	Hardening Your Security	175
	Configuration Password	175
	Creating Configuration Directory Credentials	175
	Message Queue Client Certificate Validation	176

Message Queue Self-Signed SSL Certificate	177
Access to the Message Queue Broker	177
Configuration Directory Certificate Validation	177
Restricting Access to the Configuration Directory	177
Securing Replicated Configurations	178
Using idsync certinfo	180
Arguments	180
Usage	181
Enabling SSL in Directory Server	181
▼ To Enable SSL in Directory Server	182
Retrieving the CA Certificate from the Directory Server Certificate Database	183
Retrieving the CA Certificate from the Directory Server (using dsadm command on Solaris platform)	183
Enabling SSL in the Active Directory Connector	183
Retrieving an Active Directory Certificate	184
Adding Active Directory Certificates to the Connector's Certificate Database	185
Adding Active Directory Certificates to Directory Server	187
▼ To Add the Active Directory CA certificate to the Directory Server Certificate Database	187
Adding Directory Server Certificates to the Directory Server Connector	188
▼ To Add the Directory Server Certificates to the Directory Server Connector	188
 9 Understanding Audit and Error Files	 189
Understanding the Logs	189
Log Types	190
Reading the Logs	193
Configuring Your Log Files	194
▼ To Configure Logging for Your Deployment	194
Viewing Directory Source Status	196
▼ To View the Status of your Directory Sources	196
Viewing Installation and Configuration Status	197
▼ To View the Remaining Steps of the Installation and Configuration Process	197
Viewing Audit and Error Logs	198
▼ To View Your Error Logs	198
Enabling Auditing on a Windows NT Machine	199
▼ To Enable Audit Logging on Your Windows NT Machine	199

Part II	Identity Synchronization for Windows Appendixes	201
A	Using the Identity Synchronization for Windows Command Line Utilities	203
	Common Features	203
	Common Arguments to the Idsync Subcommands	203
	Entering Passwords	205
	Getting Help	206
	Using the idsync command	206
	Using certinfo	208
	Using changepw	208
	Using importcnf	210
	Using prepsds	210
	Using printstat	214
	Using resetconn	215
	Using resync	215
	Using groupsync	218
	Using accountlockout	218
	Using dspluginconfig	219
	Using startsync	219
	Using stopsync	220
	Using the forcepwchg Migration Utility	221
	▼ To Execute the forcepwchg Command line Utility	221
B	Identity Synchronization for Windows LinkUsers XML Document Sample	223
	Sample 1: linkusers-simple.cfg	223
	Sample 2: linkusers.cfg	224
C	Running Identity Synchronization for Windows Services as Non-Root on Solaris	227
	Running Services as a Non-root User	227
	▼ To Run services as a Non-root User	227
D	Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows	229
	Understanding Synchronization User List Definitions	229

Configuring Multiple Windows Domains	231
▼ To Configure Multiple Windows Domains	232
E Identity Synchronization for Windows Installation Notes for Replicated Environments	235
Configuring Replication	235
▼ To Configure any Replication Topology	236
Configuring Replication Over SSL	237
▼ To Configure Directory Servers Involved in Replication so that all Replication Operations Occur Over an SSL Connection	237
Configuring Identity Synchronization for Windows in an MMR Environment	238
▼ To Configure Identity Synchronization for Windows in an MMR Environment	238
Index	239

Preface

This guide covers installation and configuration information for Sun Java System Identity Synchronization for Windows.

Who Should Use This Book

If you are installing Directory Server Enterprise Edition software for evaluation purposes only, put this guide aside for now, and see [Sun Directory Server Enterprise Edition 7.0 Evaluation Guide](#).

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

Before You Read This Book

Review pertinent information in the [Sun Directory Server Enterprise Edition 7.0 Release Notes](#).

If you are deploying Directory Server Enterprise Edition software in production, also review pertinent information in the [Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide](#).

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

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

How This Book Is Organized

TBD

Sun Directory Server Enterprise Edition Documentation Set

This documentation set explains how to use Sun™ Directory Server Enterprise Edition to evaluate, design, deploy, and administer directory services. In addition, it shows how to develop client applications for Directory Server Enterprise Edition. The Directory Server Enterprise Edition documentation set is available at <http://docs.sun.com/coll/1819.1>.

The following table lists all the available documents.

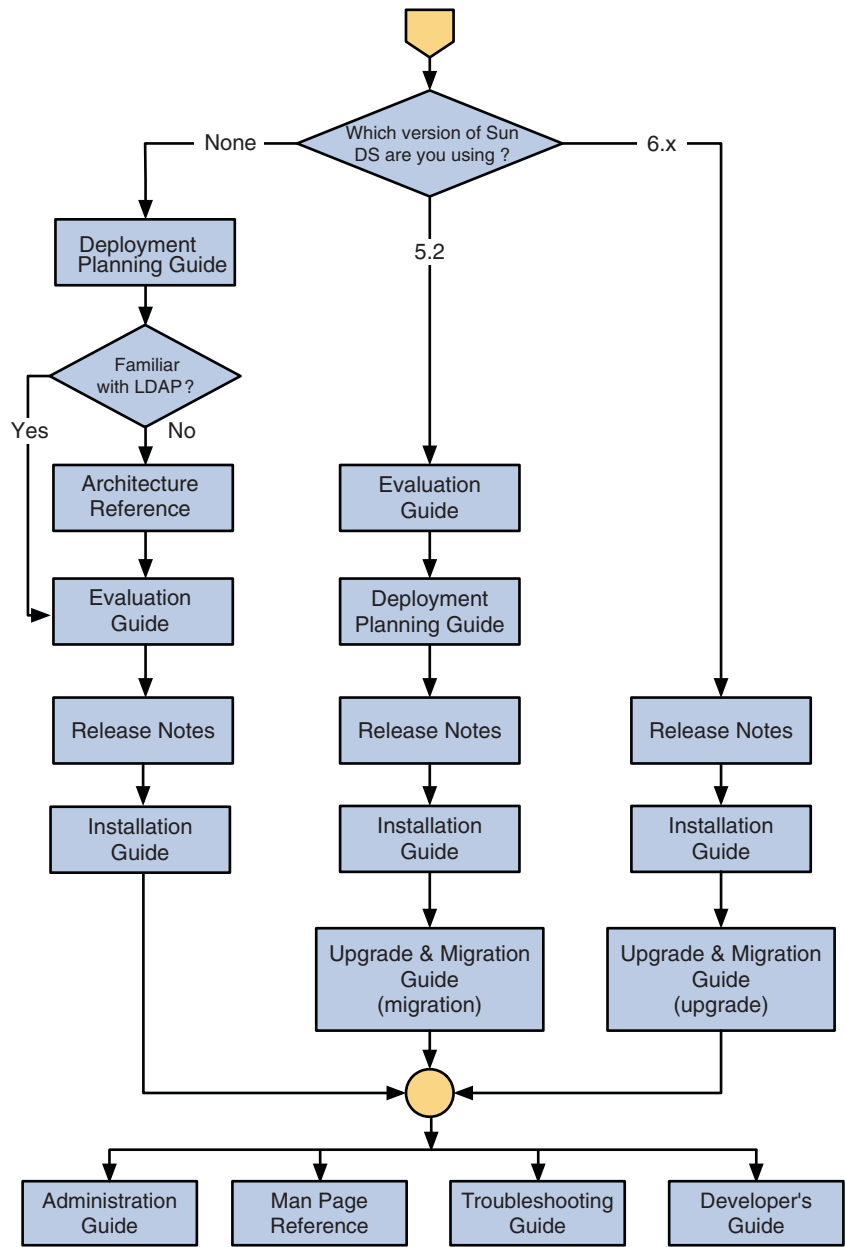
TABLE P-1 Directory Server Enterprise Edition Documentation

Document Title	Contents
<i>Sun Directory Server Enterprise Edition 7.0 Release Notes</i>	Contains the latest information about Directory Server Enterprise Edition, including known problems.
<i>Sun Directory Server Enterprise Edition 7.0 Documentation Center</i>	Contains links to key areas of the documentation set that help you to quickly locate the key information.
<i>Sun Directory Server Enterprise Edition 7.0 Evaluation Guide</i>	Introduces the key features of this release. Demonstrates how these features work and what they offer in the context of a deployment that you can implement on a single system.
<i>Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide</i>	Explains how to plan and design highly available, highly scalable directory services based on Directory Server Enterprise Edition. Presents the basic concepts and principles of deployment planning and design. Discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Directory Server Enterprise Edition.
<i>Sun Directory Server Enterprise Edition 7.0 Installation Guide</i>	Explains how to install the Directory Server Enterprise Edition software. Shows how to configure the installed software and verify the configured software.
<i>Sun Directory Server Enterprise Edition 7.0 Upgrade and Migration Guide</i>	Provides upgrade instructions to upgrade the version 6 installation and migration instructions to migrate version 5.2 installations.
<i>Sun Directory Server Enterprise Edition 7.0 Administration Guide</i>	<p>Provides command-line instructions for administering Directory Server Enterprise Edition.</p> <p>For hints and instructions about using the Directory Service Control Center, DSCC, to administer Directory Server Enterprise Edition, see the online help provided in DSCC.</p>
<i>Sun Directory Server Enterprise Edition 7.0 Developer's Guide</i>	Shows how to develop directory client applications with the tools and APIs that are provided as part of Directory Server Enterprise Edition.

TABLE P-1 Directory Server Enterprise Edition Documentation (Continued)

Document Title	Contents
<i>Sun Directory Server Enterprise Edition 7.0 Reference</i>	Introduces technical and conceptual foundations of Directory Server Enterprise Edition. Describes its components, architecture, processes, and features.
<i>Sun Directory Server Enterprise Edition 7.0 Man Page Reference</i>	Describes the command-line tools, schema objects, and other public interfaces that are available through Directory Server Enterprise Edition. Individual sections of this document can be installed as online manual pages.
<i>Sun Directory Server Enterprise Edition 7.0 Troubleshooting Guide</i>	Provides information for defining the scope of the problem, gathering data, and troubleshooting the problem areas by using various tools.
<i>Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide</i>	Provides general guidelines and best practices for planning and deploying Identity Synchronization for Windows.
<i>Sun Java System Identity Synchronization for Windows 6.0 Installation and Configuration Guide</i>	Describes how to install and configure Identity Synchronization for Windows.
Additional Installation Instructions for Sun Java System Identity Synchronization for Windows 6.0	Provides additional installation instructions in context of Directory Server Enterprise Edition 7.0.

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



Related Reading

The SLAMD Distributed Load Generation Engine is a Java™ application that is designed to stress test and analyze the performance of network-based applications. This application was

originally developed by Sun Microsystems, Inc. to benchmark and analyze the performance of LDAP directory servers. SLAMD is available as an open source application under the Sun Public License, an OSI-approved open source license. To obtain information about SLAMD, go to <http://www.slamd.com/>. SLAMD is also available as a java.net project. See <https://slamd.dev.java.net/>.

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

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

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

Identity Synchronization for Windows works with Microsoft Windows password policies.

- Information about password policies for Windows 2003, is available in the [Microsoft documentation](#) online.
- Information about the Microsoft Certificate Services Enterprise Root certificate authority, is available in the [Microsoft support documentation](#) online.
- Information about configuring LDAP over SSL on Microsoft systems, is available in the [Microsoft support documentation](#) online.

Redistributable Files

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

Default Paths and Command Locations

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

Default Paths

The table in this section describes the default paths that are used in this document. For complete descriptions of the files installed, see [Chapter 1, “Directory Server Enterprise Edition File Reference,”](#) in *Sun Directory Server Enterprise Edition 7.0 Reference*.

TABLE P-2 Default Paths

Placeholder	Description	Default Value
<i>install-path</i>	Represents the base installation directory for Directory Server Enterprise Edition software.	When you install from a zip distribution using unzip, the <i>install-path</i> is the <i>current-directory/dsee7</i> . When you install from a native package distribution, the default <i>install-path</i> is <i>/opt/SUNWdsee7</i> .
<i>instance-path</i>	Represents the full path to an instance of Directory Server or Directory Proxy Server. Documentation uses <i>/local/dsInst/</i> for Directory Server and <i>/local/dps/</i> for Directory Proxy Server.	No default path exists. Instance paths must nevertheless always be found on a <i>local</i> file system. On Solaris systems, the <i>/var</i> directory is recommended:
<i>serverroot</i>	Represents the parent directory of the Identity Synchronization for Windows installation location	Depends on your installation. Note that the concept of a <i>serverroot</i> no longer exists for Directory Server and Directory Proxy Server.
<i>isw-hostname</i>	Represents the Identity Synchronization for Windows instance directory	Depends on your installation
<i>/path/to/cert8.db</i>	Represents the default path and file name of the client's certificate database for Identity Synchronization for Windows	<i>current-working-dir/cert8.db</i>
<i>serverroot/isw-hostname/logs/</i>	Represents the default path to the Identity Synchronization for Windows local log files for the System Manager, each connector, and the Central Logger	Depends on your installation
<i>serverroot/isw-hostname/logs/central/</i>	Represents the default path to the Identity Synchronization for Windows central log files	Depends on your installation

Command Locations

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

TABLE P-3 Command Locations

Command	Native Package Distribution	Zip Distribution
cacoadm	/usr/sbin/cacoadm	Solaris, Linux, HP—UX — <i>install-path/bin/cacoadm</i>
		Windows - <i>install-path\bin\cacoadm.bat</i>
certutil	/usr/sfw/bin/certutil	<i>install-path/bin/certutil</i>
dpadm(1M)	<i>install-path/bin/dpadm</i>	<i>install-path/bin/dpadm</i>
dpconf(1M)	<i>install-path/bin/dpconf</i>	<i>install-path/bin/dpconf</i>
dsadm(1M)	<i>install-path/bin/dsadm</i>	<i>install-path/bin/dsadm</i>
dsccon(1M)	<i>install-path/bin/dsccon</i>	<i>install-path/bin/dsccon</i>
dsccreg(1M)	<i>install-path/bin/dsccreg</i>	<i>install-path/bin/dsccreg</i>
dscsetup(1M)	<i>install-path/bin/dscsetup</i>	<i>install-path/bin/dscsetup</i>
dsconf(1M)	<i>install-path/bin/dsconf</i>	<i>install-path/bin/dsconf</i>
dsmig(1M)	<i>install-path/bin/dsmig</i>	<i>install-path/bin/dsmig</i>
dsutil(1M)	<i>install-path/bin/dsutil</i>	<i>install-path/bin/dsutil</i>
entrycmp(1)	<i>install-path/bin/entrycmp</i>	<i>install-path/bin/entrycmp</i>
fildif(1)	<i>install-path/bin/fildif</i>	<i>install-path/bin/fildif</i>
idsktune(1M)	Not provided	At the root of the unzipped zip distribution
insync(1)	<i>install-path/bin/insync</i>	<i>install-path/bin/insync</i>
ldapsearch(1)	/opt/SUNWdsee/dsee6/bin	<i>install-path/dsrk/bin</i>
repldisc(1)	<i>install-path/bin/repldisc</i>	<i>install-path/bin/repldisc</i>

Typographic Conventions

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

TABLE P-4 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name%</code> su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX® system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-5 Shell Prompts

Shell	Prompt
C shell	<code>machine_name%</code>
C shell for superuser	<code>machine_name#</code>
Bourne shell and Korn shell	<code>\$</code>
Bourne shell and Korn shell for superuser	<code>#</code>

Shell Prompts in Command Examples

The following table shows default system prompts and superuser prompts.

TABLE P-6 Shell Prompts

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

Symbol Conventions

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

TABLE P-7 Symbol Conventions

Symbol	Description	Example	Meaning
[]	Contains optional arguments and command options.	ls [-l]	The -l option is not required.
{ }	Contains a set of choices for a required command option.	-d {y n}	The -d option requires that you use either the y argument or the n argument.
\${ }	Indicates a variable reference.	\${com.sun.javaRoot}	References the value of the com.sun.javaRoot variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Documentation, Support, and Training

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

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

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 Feedback.



PART I

Installing Identity Synchronization for Windows

Sun Java System Identity Synchronization for Windows allows passwords and other specified user attributes to flow between Sun Java System Directory Server and other systems.

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

For the latest information about new features and about enhancements in this release of Identity Synchronization for Windows, see the [Sun Directory Server Enterprise Edition 7.0 Release Notes](#).

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

This part includes the following chapters:

- [Chapter 1, “Understanding the Product,”](#) describes Identity Synchronization for Windows product features, system components and their distribution, command-line utilities, and deployment examples.
- [Chapter 2, “Preparing for Installation,”](#) describes the installation and configuration processes and information you need to know when preparing to install the product.
- [Chapter 3, “Installing Core,”](#) explains how to use the Identity Synchronization for Windows installation program and how to install its Core component.
- [Chapter 4, “Configuring Core Resources,”](#) explains how to add and configure Core resources by using the Console.
- [Chapter 5, “Installing Connectors,”](#) provides instructions for installing the Identity Synchronization for Windows Connectors and Directory Server Plug-ins.
- [Chapter 6, “Synchronizing Existing Users and User Groups,”](#) explains how to link and resynchronize existing users and user groups for new Identity Synchronization for Windows installations.
- [Chapter 7, “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 8, “Configuring Security,”](#) describes how to configure a secure system. This chapter covers how to harden security, secure replicated configurations, enable SSL, and add Active Directory CA certificates to certificate databases.
- [Chapter 9, “Understanding Audit and Error Files,”](#) provides information about audit and error logging, including instructions on how to set logging levels, how to view and understand your log files, and directory source status.
- [Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities,”](#) shows how to use command-line utilities to perform various tasks.
- [Appendix B, “Identity Synchronization for Windows LinkUsers XML Document Sample,”](#) provides sample Linkusers XML configuration files that you can use to customize your deployment.
- [Appendix C, “Running Identity Synchronization for Windows Services as Non-Root on Solaris,”](#) explains how to run Identity Synchronization for Windows services as a non-root user on the Solaris operating system.
- [Appendix D, “Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows,”](#) provides information about Synchronization User List definitions and multiple domain configurations.
- [Appendix E, “Identity Synchronization for Windows Installation Notes for Replicated Environments,”](#) provides an overview of the steps required to configure and secure a multimaster replication deployment.

Understanding the Product

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

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

Identity Synchronization for Windows 6.0 supports Sun Directory Server 7.0, 6.3, 6.2, 6.1, 6.0, and 5.2 Patch 5.

Sun Java System Identity Synchronization for Windows handles synchronization events in these ways:

- **Securely.** It does not send passwords “in the clear,” and it restricts system access to administrators only.
- **Robustly.** It keeps directories synchronized, even when individual components are temporarily unavailable.
- **Efficiently.** It uses synchronization methods that place very little load on your directory servers.

Note – Before you install Sun Java System Identity Synchronization for Windows version 6.0, you *must* read the [Technical Note](#). This Technical Note provides additional installation instructions that help you to install Identity Synchronization for Windows for Directory Server Enterprise Edition 7.0.

Sun Java System Identity Synchronization for Windows version 6.0 is not bundled with the Sun Directory Server Enterprise Edition 7.0 release. You can download the Identity Synchronization for Windows software from http://www.sun.com/software/products/directory_srvr_ee/get.jsp.

You should also familiarize yourself with the concepts described in this chapter, which includes the following topics:

- [“Product Features” on page 24](#)
- [“System Components” on page 25](#)
- [“System Components Distribution” on page 31](#)
- [“How Identity Synchronization for Windows Detects Changes in Directory Sources” on page 34](#)
- [“Deployment Example: A Two-Machine Configuration” on page 40](#)

Product Features

Sun Java System Identity Synchronization for Windows provides the following features and functionality:

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

Synchronizing passwords allows users to access applications using these directory sources for login authentication, so users 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 location.

- **Bidirectional user attributes synchronization.** Enables you to create, modify, and delete selected attributes in one directory environment and propagate the values automatically to the other directory environment.
- **Bidirectional user account creation synchronization.** Enables you to create or delete a user account in one directory environment and automatically propagate the new account to the other directory environment.
- **Bidirectional group synchronization.** Enables you to synchronize the creation or deletion of a group, and association or disassociation of users with that group between Directory Server and Active Directory sources.
- **Bidirectional object deletions, activations, and inactivations.** Enable you to control the flow of object deletions, activations, and inactivations between Directory Server and Active Directory sources.
- **Bidirectional account lockout and unlockout synchronization.** Enables you to synchronize account lockout and unlockout between Directory Server and Active Directory sources.
- **Synchronization with multiple domains.** Enables you to synchronize with multiple Active Directory and Windows NT domains, and with multiple Active Directory forests.

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

You are not 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 do not need to install any components in the Windows operating system.

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

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

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

System Components

The following figure shows that Identity Synchronization for Windows consists of a set of Core components and any number of individual connectors and connector subcomponents. These system components allow for the synchronization of password and user attribute updates between Sun Java System Directory Server (Directory Server) and Windows directories.

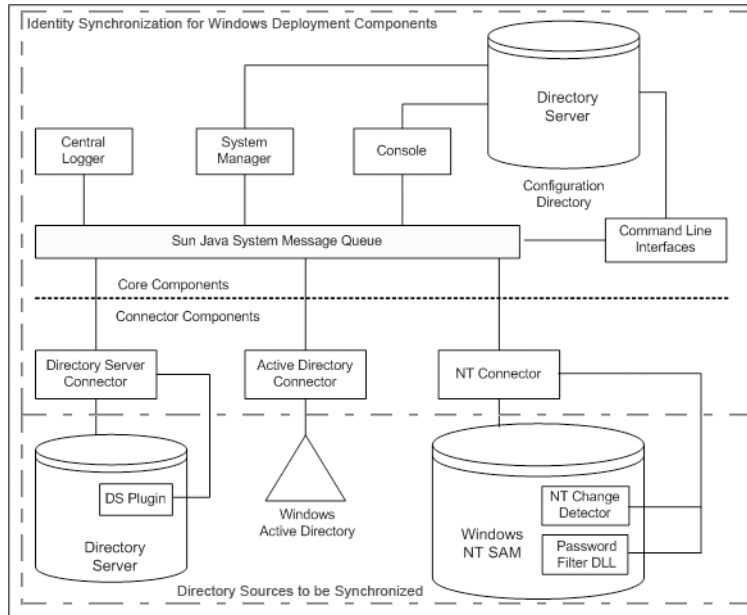


FIGURE 1-1 System Components

This section defines and describes these Identity Synchronization for Windows components:

- [“Watchdog Process” on page 26](#)
- [“Core” on page 27](#)
- [“Connectors” on page 29](#)
- [“Connector Subcomponents” on page 30](#)
- [“Message Queue” on page 31](#)

Watchdog Process

The *Watchdog* is an Identity Synchronization for Windows Java technology-based process (Java process) that starts, restarts, and stops individual background Java processes. The Watchdog launches and monitors the central logger, system manager, and connectors. The Watchdog does not monitor subcomponents, Message Queue, or the Identity Synchronization for Windows Console.

The Watchdog is installed where you install the Core components and it can be started as a Solaris™ software daemon, Red Hat Linux daemon, or a Windows service.

Core

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

The Core component consists of the following components:

- “Configuration Directory” on page 27
- “Console” on page 27
- “Command-Line Utilities” on page 28
- “System Manager” on page 28
- “Central Logger” on page 28

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 utilities, and the installer all read and write the product’s configuration data to and from the configuration directory, including the following:

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

Console

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

You can use the Console to do the following:

- Configure directory sources to be synchronized
- Define mappings for user entry attributes to be synchronized, in addition to passwords
- Specify which users and attributes within a directory or domain topology will 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 Secure Sockets Layer (SSL) settings
- Change the Identity Synchronization for Windows configuration password
- Configure the Directory Server Plug-in for a specified Directory Server source
- Prepare a Sun Java System Directory Server source for use by Identity Synchronization for Windows
- Display the steps that you must perform to complete the installation or configuration process, and view the status of installed connectors, the system manager, and Message Queue
- Reset connector states in the configuration directory to *uninstalled*
- Synchronize and link existing users in two directories, and pre-populate directories as part of the installation process
- Enable or disable account lockout
- Enable or disable group synchronization
- Start and stop synchronization

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

System Manager

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

- Leverages the product's back-end networked facilities to dynamically deliver configuration updates to connectors
- Keeps the 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, having all logging information centralized is of great administrative value. This centralization 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 perform these tasks:

- 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 sources

The two types of logs are as follows:

- **Audit log.** Provides information about the system's day-to-day activities, which includes 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.
- **Error log.** Provides information about conditions that are 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.

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

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. These connectors bidirectionally synchronize 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 source. You can use multiple connectors for additional domains.
- **Windows NT Connector.** Supports a single domain on Windows NT.

Note – The Watchdog is installed where you install a connector, and it starts, restarts, and stops the connectors. For more information, see [“Watchdog Process” on page 26](#).

Connector Subcomponents

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

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

- [“Directory Server Plug-In” on page 30](#)
- [“Windows NT Connector Subcomponents” on page 30](#)

Note – Active Directory Connectors do not require subcomponents.

Directory Server Plug-In

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

This Plug-in does the following:

- 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 37](#))

Note – Identity Synchronization for Windows used to support only two-way multimaster replication (MMR). Now, the Directory Server Plug-in is also functional in N-way MMR environments.

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, then passes the changes to the Connector
- **Password Filter DLL.** Captures password changes made on the Windows NT Domain Controller and passes these securely to the NT Connector.

Message Queue

Identity Synchronization for Windows uses Sun Java SystemMessage Queue (Message Queue), a persistent message queue mechanism with a publish and subscribe model, to propagate attribute and password changes between directory sources. Message Queue also distributes 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 open standard. This 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 service is composed of one or more dedicated message brokers that control access to the message queue, maintain information about active publishers and subscribers, and ensure that messages are delivered.

Message Queue does the following:

- 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 discusses the following:

- [“Core” on page 32](#)
- [“Directory Server Connector and Plug-in” on page 32](#)
- [“Active Directory Connector” on page 32](#)
- [“Windows NT Connector and Subcomponents” on page 33](#)

When you understand the basic concepts described in this section and in [“Deployment Example: A Two-Machine Configuration” on page 40](#), you should be able to extrapolate the information to create deployment strategies for more complex, sophisticated scenarios. Such scenarios might be mixed Active Directory and Windows NT environments or multiserver environments.

Core

Note – Install Sun Java System Message Queue 3.6 Enterprise Edition on the same machine where you are planning to install Core.

Install all Core components only once in any of the supported operating system's directory servers. Identity Synchronization for Windows installs Administration Server on your machine if it is not already installed.

Directory Server Connector and Plug-in

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

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

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

Active Directory Connector

You can install Active Directory Connectors on any of the supported operating systems. You are not required to install an Active Directory Connector on a machine running Windows. However, one Active Directory Connector must be installed for each Active Directory domain. See the following figure for a sample distribution of components.

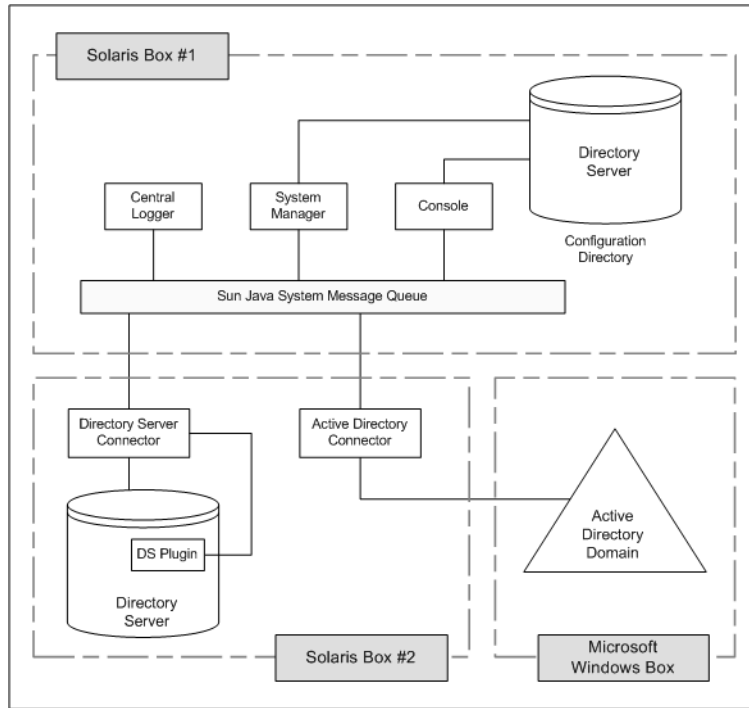


FIGURE 1-2 Directory Server and Active Directory Component Distribution

Windows NT Connector and Subcomponents

To synchronize with Windows NT SAM Registries, you must install the Windows NT Connector in the Primary Domain Controller (PDC). The installation program also 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. See the following figure for a sample distribution of components.

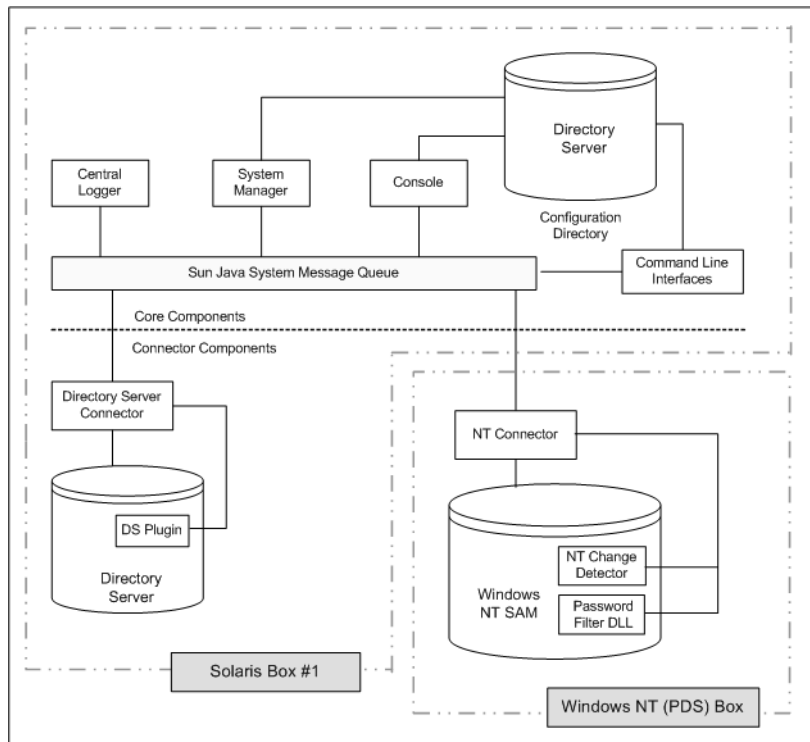


FIGURE 1-3 Directory Server and Windows NT Component Distribution

How Identity Synchronization for Windows Detects Changes in Directory Sources

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

The information is organized as follows:

- [“How Directory Server Connectors Detect Changes” on page 35](#)
- [“How Active Directory Connectors Detect Changes” on page 35](#)
- [“How Windows NT Connectors Detect Changes” on page 36](#)
- [“Propagating Password Updates” on page 37](#)
- [“Reliable Synchronization” on page 39](#)

How Directory Server Connectors Detect Changes

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

For more information about retro changelog, see [“Replication and the Retro Change Log Plug-In” in *Sun Directory Server Enterprise Edition 7.0 Reference*](#).

- Capture clear-text passwords by encrypting them and then making them available in the retro changelog. Without the Plug-in, only hashed passwords appear in the retro changelog, and hashed passwords cannot be synchronized.
- Perform on-demand password synchronization with Active Directory. No Identity Synchronization for Windows components need to be installed in a Windows topology (See [“Using On-Demand Password Synchronization to Obtain Clear-Text Passwords” on page 37](#)).

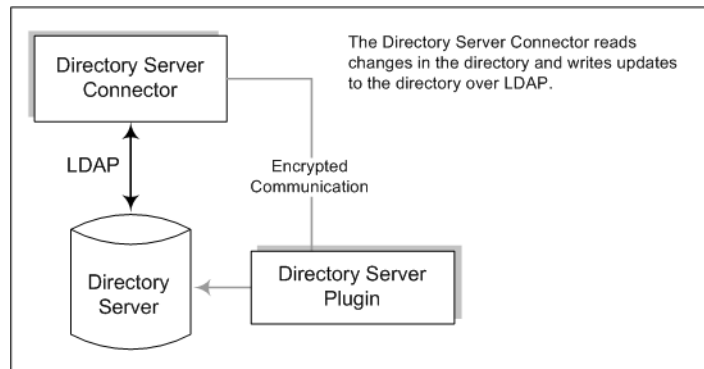


FIGURE 1-4 How Directory Server Connectors Detect Changes

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, an Active Directory Connector uses 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 on Windows. These connectors can also run on other operating systems such as Solaris or Red Hat Linux, and detect or make changes remotely over LDAP.

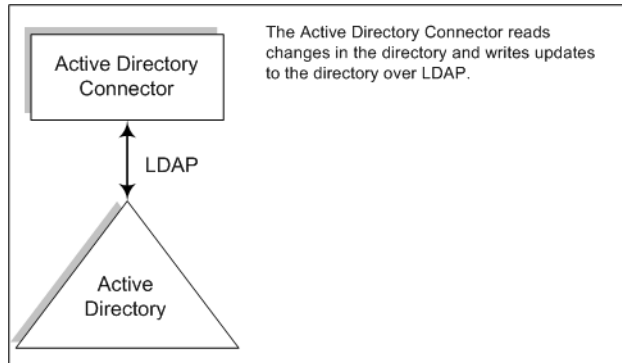


FIGURE 1-5 How Active Directory Connectors Detect Changes

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. Auditing must be enabled or Identity Synchronization for Windows cannot read log messages from Windows NT machine. To verify that audit logging is enabled, see [“Enabling Auditing on a Windows NT Machine”](#) on page 199.

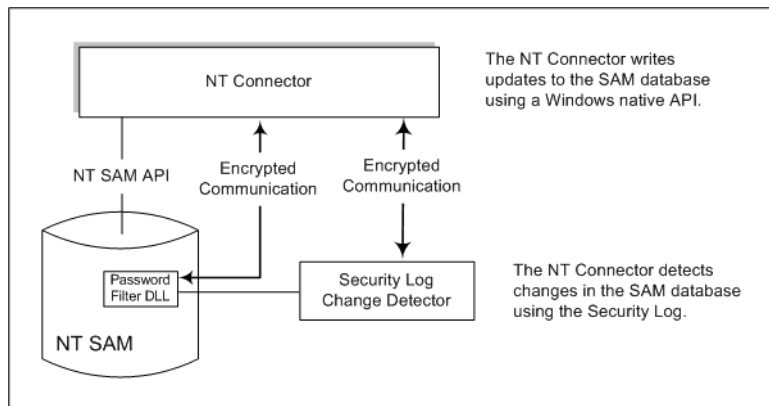


FIGURE 1-6 How Windows NT Connectors Detect Changes

For a description of the Change Detector and the Password Filter DLL subcomponents, see [“Windows NT Connector Subcomponents” on page 30](#).

Propagating Password Updates

This section explains two ways to obtain clear-text passwords. Clear-text passwords are needed to propagate password changes between Windows and Directory Server sources.

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, the passwords have already been encrypted.

Windows NT provides a Password Filter DLL interface that 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 (not the Primary Domain Controller used by Window NT). 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.

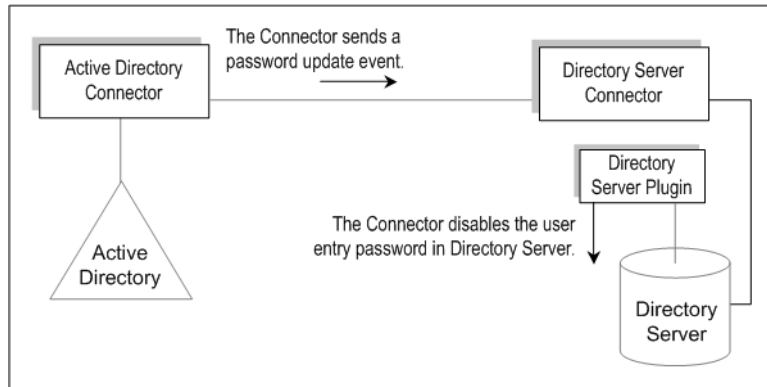
On-demand password synchronization provides a method to obtain new password values on Directory Server when users try to login after their password change on Windows 2000/2003.

On-demand password synchronization also allows you to synchronize passwords on Active Directory without using the Password Filter DLL.

The on-demand password synchronization process is as follows:

1. The user presses Ctrl-Alt-Del on a machine running Windows and changes his or her password. The new passwords are stored in Active Directory.
2. The Active Directory Connector polls the system at scheduled intervals.

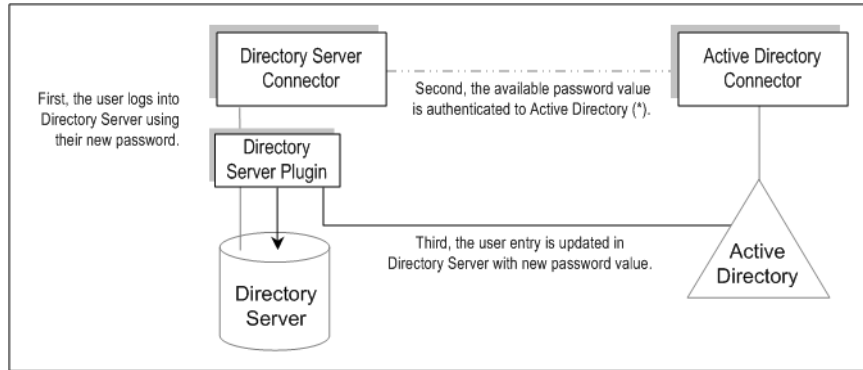
When the Connector detects the password change, based on changes made to the `USNchanged` (Update Sequence Number) and `PwdLastSet` attributes, the Connector publishes a message on Message Queue about the password change. The message is transferred on an SSL-encrypted channel.



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 `dspswval` attribute to `true`, which invalidates the old password and alerts the Directory Server Plug-in of the password change.
5. When the user tries to log in, using an LDAP application (such as Portal Server) to authenticate against the Directory Server, the Sun Java System Directory Server Plug-in detects that the password value in the Directory Server entry is invalid.
6. The Directory Server Plug-in searches for the corresponding user in Active Directory. When the Plug-in finds the user, the Plug-in tries to bind to Active Directory using the password provided when the user tried logging in to Directory Server.

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

7. If the bind against Active Directory succeeds, the Directory Server Plug-in sets the password and removes the invalid password flag from the user entry on Directory Server allowing the user to log in.



Note – If user authentication fails, the user entry password remains in Directory Server and the passwords on Directory Server and Active Directory are not the same 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 loosely 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 after 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 and contains all changes, even to passwords. This version of Identity Synchronization for Windows does not support Sun™ Cluster software or other true high-availability solutions. Because users do not interact with Identity Synchronization for Windows directly, high availability is not usually required. If you experience a catastrophic failure, you can reinstall Identity Synchronization for Windows components and use the `idsync resync` command to resynchronize 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 Servers.
If the preferred Directory Server is unavailable, the Directory Server Connector will apply changes to one of the available secondary servers from the MMR topology.
- While the Active Directory Connector can communicate with a single Active Directory domain controller only, the Directory Server Plug-in can fail between all Active Directory domain controllers while performing on-demand password synchronization. This point is where failover is most important. If the Directory Server Plug-in cannot contact an Active Directory domain controller to verify a user's new password, the user cannot log in to Directory Server.

Deployment Example: A Two-Machine Configuration

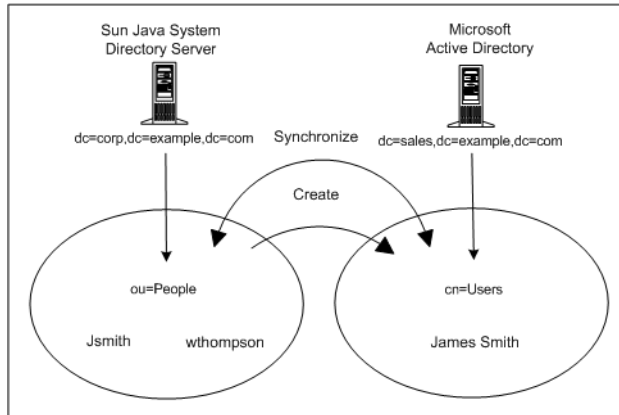
This section describes a deployment scenario in which Identity Synchronization for Windows is used to synchronize user object creation and bidirectional password modification operations between Directory Server and Active Directory sources.

The deployment scenario consists of two machines:

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

Note – Even though Windows NT is not used in this scenario, Identity Synchronization for Windows also supports synchronization with NT domains.

The following figure illustrates the synchronization requirements (node structures with associated attribute values) used for this deployment scenario.



The two goals for this scenario are as follows:

- 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 in the Directory Server, the new password should automatically be synchronized to cn=James Smith in the cn=users container in Active Directory.

- 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, Identity Synchronization for Windows will create a new account cn=William Thompson in the cn=Users container with the same set of attributes in Active Directory.

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

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

Physical Deployment

The following figure illustrates how all the product's components are physically deployed on a single Solaris system, while the Active Directory domain resides in a separate Active Directory domain controller where no components have been installed.

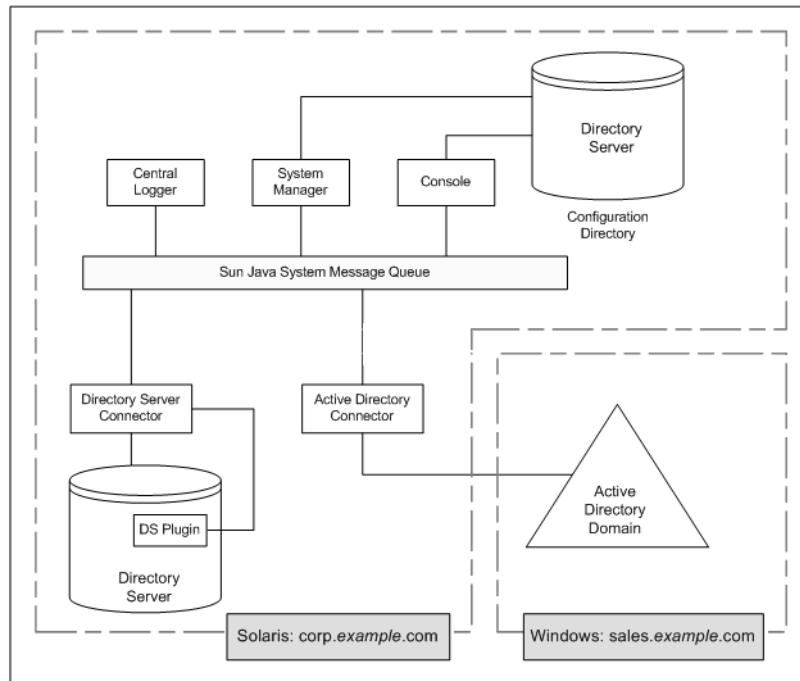


FIGURE 1-7 Directory Server and Active Directory Scenario

Component Distribution

corp.example.com is a machine where Directory Server is installed on a Solaris operating system. The root suffix for the Directory Server instance being synchronized is dc=corp,dc=example,dc=com.

This topology contains the following:

- Identity Synchronization for Windows Core components
- Identity Synchronization for Windows Directory Server Connector
- Identity Synchronization for Windows Directory Server Plug-in

- Identity Synchronization for Windows configuration directory (located in a different Directory Server instance than the one being synchronized)
sales.example.com is the Active Directory domain being synchronized.

Preparing for Installation

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

For information about the Identity Synchronization for Windows installation requirements, see [Chapter 6, “Identity Synchronization for Windows Bugs Fixed and Known Problems,”](#) in *Sun Directory Server Enterprise Edition 7.0 Release Notes*.

Identity Synchronization for Windows can also be installed in French, German, Spanish, Japanese, Korean, Simplified Chinese, and Traditional Chinese languages. All the languages are bundled in the same distribution.

For multilingual support for Identity Synchronization for Windows, use the UTF-8 encoding.

This chapter covers the following topics:

- “Installation Overview” on page 45
- “Configuration Overview” on page 49
- “Synchronizing Passwords With Active Directory” on page 53
- “Configuring Windows for SSL Operation” on page 59
- “Installation and Configuration Decisions” on page 60
- “Installation Checklists” on page 63

Installation Overview

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

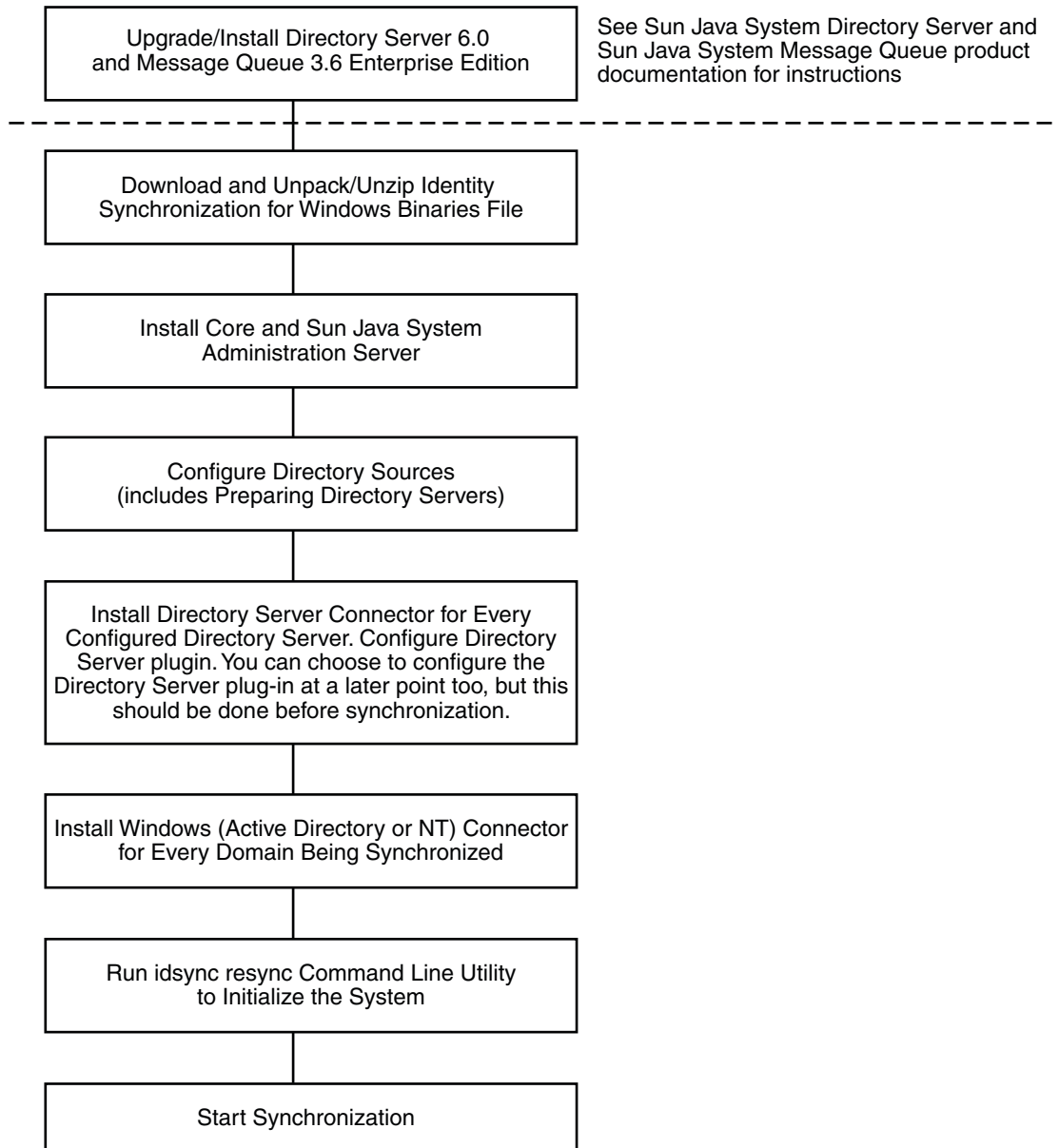


FIGURE 2-1 Single-host installation procedure

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

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

This is a list of remaining installation and configuration steps:

- ✓ 1 : Install the Identity Synchronization core components.
- 2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importenf'.
- 3 : Prepare every Sun Directory Server included in this configuration by using the console or the 'idsync prepds' command.
- 4 : Install connectors for every configured directory source.
- 5 : After installing each Sun Directory Server connector, configure the Sun Directory Server plugin on every master and on every read-only replica by using the console or the 'idsync dspluginconfig' command.
- 6 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.
- 7 : Start synchronization using the console or the 'idsync startsync' command.

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

As you go through the installation and configuration process, all completed steps in the list are grayed-out as shown in Figure 6-2.

The rest of this section provides an overview of the installation and configuration process.

Installing Core

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

- **Sun Java System Administration Server.** Configures the Directory Server Plug-in and provides the administration framework.
- **Console.** Provides a centralized location for performing all of the product's component configuration and administration tasks.
- **Central logger.** Centralizes all audit and error logging information in a central location.
- **System manager.** Delivers configuration updates to connectors dynamically and maintains the status of each connector.
- Instructions for installing Core are provided in [Chapter 3, “Installing Core”](#)

Configuring the Product

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

Instructions for configuring directory resources are provided in [Chapter 4, “Configuring Core Resources.”](#)

Preparing the Directory Server

Before you can install Directory Server Connectors, you must prepare a Sun Java System Directory Server source for every preferred and secondary Directory Server that is being synchronized.

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

Instructions for preparing Directory Server are provided in [“Preparing Sun Directory Source” on page 91](#).

Installing Connectors and Configuring Directory Server Plug-In

You can install any number of connectors depending on the number of configured directories in your topology. Both the Console and the installation program use the directory label to associate a connector with the directory that is synchronized. The following table describes the label naming conventions.

TABLE 2-1 Label Naming Conventions

Connector Type	Directory Source Label	Subcomponent
Directory Server Connector	root suffix or suffix/database	Directory Server Plug-in Configure one Plug-in in every Directory Server (master or consumer) for the root suffix being synchronized.
AD Connector	Domain name	None
NT Connector	Domain name	(Automatically installed with the Windows NT Connector) Change Detector and Password Filter DLL subcomponents are installed together in the same installation. You must install the Windows NT Connector using the graphical user interface (GUI) installer.

TABLE 2-2 Label Naming Examples

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

Instructions for installing and configuring Connectors are provided in [Chapter 3, “Installing Core”](#)

Synchronizing Existing Users

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

- Link existing entries (for more information about *linking users*, see [“Linking Users” on page 153](#))
- 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

Instructions for synchronizing existing users in your deployment are provided in [Chapter 6, “Synchronizing Existing Users and User Groups.”](#)

Configuration Overview

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

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

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

- Directories
- Synchronization Settings
- Object classes
- Attributes and Attribute Mapping
- Synchronization User Lists

Note – Some related configuration instructions appear in [Chapter 4, “Configuring Core Resources.”](#)

Directories

A directory represents the following:

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

You can configure any number of each directory type.

Synchronization Settings

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

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

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

Object Classes

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

Note – Object classes are not applicable for Windows NT.

Identity Synchronization for Windows supports two types of object classes:

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

For instructions on configuring object classes 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 and one or more values, and follows a standard syntax for the type of information that can be stored as the attribute value.

You can define attributes from the Console. See [Chapter 4, “Configuring Core Resources.”](#)

Attribute Types

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

- **Significant attributes.** Synchronized between Directory Server and Windows directories whenever the attributes are modified according to specified modification synchronization settings.
- **Creation attributes.** Synchronized between Directory Server 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” 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 Directory Server side, if you are configuring `inetorgperson` of a user object class, 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 52](#))

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, follow these guidelines:

- You can use multiple attributes in a creation expression (`cn=%givenName%.%sn%`), but the attributes in *%attribute_name%* must have single values.
- If `A=0`, `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, map the attribute names between the Directory Server and Active Directory/Windows NT systems to synchronize them to each other. For example, you must map the Sun `inetorgperson` attribute to the Active Directory `user` attribute.

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 Directory Server 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?

See [Appendix D, “Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows,”](#) for detailed information about creating SULs.

Synchronizing Passwords With Active Directory

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

Identity Synchronization for Windows services must occasionally create entries that do not have passwords, for example, during a `resync -c` from Directory Server to Active Directory. Consequently, if password policies are 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 need to understand the issues associated with enforcing their password policies.

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 the Solaris OS, Red Hat Linux, or Windows.

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

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

Enforcing Password Policies

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

If you create users on Active Directory (or Directory Server) that meet the required password policies for that topology, the users may be created and synchronized properly between the two systems. If you have password policies enabled on both directory sources, the passwords must meet the policies of both directory sources 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 in both Active Directory and Directory Server, you should enable password policies in the directory source that you consider to be the authoritative source for passwords and user creations. However, user creations will not work as expected in some cases because of certain password policy configurations.

Note – Identity Synchronization for Windows does not synchronize password expiration.

This section discusses the following:

- [“Directory Server Password Policies” on page 54](#)
- [“Active Directory Password Policies” on page 55](#)
- [“Creating Accounts Without Passwords” on page 55](#)
- [“Example Password Policies” on page 58](#)
- [“Error Messages” on page 59](#)

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

in to Directory Server, which triggers on-demand password synchronization. At this time the login will fail because the password violates the Directory Server password policy.

To recover from this situation, do one of the following:

- Force users to change their password the next time they log in to Active Directory.
- Change the user password in Active Directory, making sure that the new password meets Directory Server password policy requirements.

Active Directory Password Policies

If you create users in Active Directory that do not match the Active Directory password policy, those users *will* be created in 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 in Directory Server. The users will not have a password in Directory Server, so no one will be able to log in as those users. 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, the temporarily created users will be deleted automatically.
- Users are created without a password in Directory Server. Directory Server does not enforce the password policy for user creations unless the entries contain a password.

The preferred method from recovering this situation is to synchronize deletions from Active Directory to Directory Server. Alternatively, you can remove the users from Directory Server and then add them to Active Directory with a password that follows Active Directory password policies. This method ensures that the users are created in Directory Server and are properly linked. Directory Server users will have their password invalidated when they log in to Active Directory for the first time and change it.

- 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 in Directory Server. The entries will not be linked, and you will have to run the `idsync resync` command to link the two separate accounts.

If you run the `idsync resync` command, you must reset the passwords for the accounts in Active Directory that were linked to entries in Directory Server. Resetting the passwords invalidates those passwords in Directory Server, which then forces on-demand synchronization to update the Directory Server passwords the next time users authenticate 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 Directory Server without a password, it sets the userpassword attribute to {PSWSYNC}*INVALID*PASSWORD*. The user will not be able to log in to Directory Server until you reset the password. One exception 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 Active Directory without a password, it sets the user’s password to a randomly chosen, strong password that meets Active Directory password policies. In this case, a warning message is logged, and the user will not be able to log in to Active Directory until you reset the password.

The following tables show some scenarios that you might encounter as you work with Identity Synchronization for Windows.

This section describes how password policies affect synchronization and resynchronization.

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

TABLE 2–3 How Password Policies Affect Synchronization Behavior

Scenario			Results		
User Originally Created In	User Meets Password Policy In		User Created In		
	Directory Server	Active Directory	Directory Server	Active Directory	Comments
Active Directory	Yes	Yes	Yes	Yes	

TABLE 2-3 How Password Policies Affect Synchronization Behavior (Continued)

Scenario			Results		
User Originally Created In	User Meets Password Policy In		User Created In		
	Directory Server	Active Directory	Directory Server	Active Directory	Comments
	Yes	No	Yes (see Comments)	No	User will be created in Directory Server. However, if deletions are synchronized from Active Directory to Directory Server, this user will be deleted immediately. See “ Active Directory Password Policies ” on page 55 information.
	No	Yes	Yes	Yes	See “ Active Directory Password Policies ” on page 55 information.
	No	No	Yes (see Comments)	No	Users will be created in Directory Server. However, if deletions are synchronized from Active Directory to Directory Server, this user will be deleted immediately. See “ Active Directory Password Policies ” on page 55 information.
Directory Server	Yes	Yes	Yes	Yes	
	Yes	No	Yes	No	
	No	Yes	No	No	
	No	No	No	No	

TABLE 2-4 How Password Policies Affect Resynchronization Behavior

Scenario			
Resync Command	User Meets Password Policy In		Result
	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 55.
	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 55.
resync -c -i NEW_USERS NEW_LINKED_USERS	Yes	N/A	User will be created in Directory Server, and the user's passwords will be set when the user first logs in. See “Creating Accounts Without Passwords” on page 55.
	No	N/A	User will be created in Directory Server but cannot log in because the password violates the Directory Server password policy. See “Creating Accounts Without Passwords” on page 55.
resync -c	Yes	N/A	User will be created in Directory Server but cannot log in until a new password value is set in Active Directory or Directory Server. See “Creating Accounts Without Passwords” on page 55.
	No	N/A	User will be created in Directory Server but cannot log in until a new password value is set in Active Directory or Directory Server. See “Creating Accounts Without Passwords” on page 55.

Example Password Policies

This section states example password policies for Active Directory and Directory Server.

Directory Server Password Policies

- 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 minimum length is 7 characters

Active Directory Password Policies

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

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://technet.microsoft.com/en-us/library/cc782657\(ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc782657(ws.10).aspx)

For more information about password policies for Sun Java System Directory Server, see [Chapter 7, “Directory Server Password Policy,” in *Sun Directory Server Enterprise Edition 7.0 Administration Guide*](#).

Configuring Windows for SSL Operation

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

The Identity Synchronization for Windows Active Directory Connector installer can automatically setup SSL in the Active Directory Connector if you enable LDAP over SSL in Active Directory. You can automatically obtain 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 the technical note at <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 183.

Installation and Configuration Decisions

This section provides installation and configuration summaries and details the choices you make when deploying Identity Synchronization for Windows. Read all of the information in this section, and complete the installation checklists before you begin the installation process.

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. If you do, you must identify the port as an SSL port during the installation process.
- **Root suffix.** Specify the root suffix for the configuration directory. All configuration information is stored under this suffix.
- **Administrator's name and password.** Specify credentials for accessing the configuration Directory Server.
- **Configuration password.** Specify a secure password to protect sensitive configuration information.
- **File system directory.** Specify the location in which to install Identity Synchronization for Windows. You must install Core in the same directory as a Directory Server Administration Server.
- **Unused port number.** Specify an available port number for the Message Queue instance.
- **Administration Server.** Specify administration server administrator's user name and password if it already exists on Directory Server.

Core Configuration

You must provide the following information when you configure Core:

- **Sun Java System Directory schema.** Specify the Directory Server data that you want to load 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 Windows directory sources.
- **Modifications, creations, and deletions flow.** Specify how you want modifications, creations, and deletions to be propagated between Directory Server and Windows directory sources.
 - From Directory Server to Active directory/Windows NT
 - From Active directory/Windows NT to Directory Server
 - Bidirectionally

Specify whether to synchronize object activations and inactivations if they are propagated between Directory Server and Windows directory sources, 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 that stores 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 Windows NT.
- **Sun Java System Directory Servers.** Specify Directory Server instances that store users to be synchronized.

Connector Installation and Configuring the Directory Server Plug-In

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

- **Configuration directory host and port.** Specify the configuration directory host and port for the Directory Server instance on which Identity Synchronization for Windows configuration information will be stored.
- **Root suffix.** Specify the root suffix for the configuration directory. Use the root suffix specified during Core installation.
- **Administrator's name and password.** Specify credentials for accessing the configuration Directory Server.
- **Configuration password.** Specify a secure password to protect sensitive configuration information.
- **File system directory.** Specify the location in which to install Identity Synchronization for Windows. All components installed on the same machine must have the same installation path.
- **Directory sources:** Specify the directory source for which you want to install the connector or plug-in.

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

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

Using the Command-Line Utilities

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

- `certinfo` — Displays certificate information based on your configuration and SSL settings.
- `changepw` — Changes the Identity Synchronization for Windows configuration password.
- `prepds` — Prepares a Sun Java System Directory Server source for use by Identity Synchronization for Windows.
- `printstat` — Prints the status of installed connectors, the system manager, and Message Queue.

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

- `resetconn` — Resets connector states in the configuration directory to *uninstalled* only in cases of hardware or uninstaller failure.
- `resync` — Resynchronizes and links existing users, and pre-populates directories as part of the installation process.
- `dspluginconfig` — Configures or unconfigures the Directory Server Plug-in.
- `groupsync` — Enables or disables group synchronization.

- `accountlockout` — Enables or disables account lockout feature.
- `startsync` — Starts synchronization.
- `stopsync` — Stops synchronization.

See [Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities,”](#) for detailed information about these utilities.

Installation Checklists

Use these checklists to prepare for the installation process. Print the checklists and record the appropriate information before installing Identity Synchronization for Windows.

TABLE 2-5 Core Installation Checklist

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

TABLE 2-6 Core Configuration Checklist

Required Information	Entry
Active Directory global catalog (when appropriate)	
Directory Server schema server	
Directory Server user structural and auxiliary object classes	
Synchronized attributes	
Flow for user entry creations	
Flow for user entry modifications	
Flow for user entry activations and inactivations	
Flow for user entry deletions	

TABLE 2-6 Core Configuration Checklist (Continued)

Required Information	Entry
Sun Java System Directory Server directory sources	
Active Directory	
Synchronization User Lists	
Windows source filter creation expression	
Sun Java System source filter creation expression	
User name and password for the Administration Server	

Connector and Directory Server Plug-in Installation Checklist

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

Linking Users Checklist

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

Resynchronization Checklist

Required Information	Entry
Synchronization User List selection	
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?	

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” on page 67
- “Starting the Installation Program” on page 68
- “Installing Core” on page 70

Before You Begin

Before starting the Identity Synchronization for Windows installation process:

- Read [Chapter 2, “Preparing for Installation,”](#) that contains important information, such as installation prerequisites, checklists, and administrator privilege requirements.
- A Java Runtime Environment (JRE) is not provided with this product. If necessary, you can download a Java Development Kit from the following location:

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

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

Note – If Directory Server 6.x is installed with Java ES, JRE 1.5.0_09 is already installed on your computer.

- **On Windows systems only:** You must close any open Service Control Panel windows before starting Core installation, or the installation will fail.
- **On Solaris systems:** Do not install Message Queue and Identity Synchronization for Windows in the same directory.

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

Starting the Installation Program

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

- “On Solaris SPARC” on page 68
- “On Solaris x86” on page 68
- “On Windows” on page 69
- “On Red Hat Linux” on page 69

On Solaris SPARC

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

▼ To Run Identity Synchronization for Windows on Solaris SPARC

- 1 Log in as root.
- 2 Change to the directory on the delivery media for Solaris SPARC containing the installation program, `DSEE_Identity_Synchronization_for_Windows`.
- 3 Type `./runInstaller.sh` to execute the installation program.

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

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

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

On Solaris x86

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

- 1 Log in as root.
- 2 Change to the directory on the delivery media for Solaris x86 containing the installation program, `DSEE_Identity_Synchronization_for_Windows`.

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

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

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

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

On Windows

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

▼ To Run Identity Synchronization for Windows on Windows

- 1 Log in as an Administrator.**
- 2 Change to the directory on the delivery media for Windows containing the installation program, `DSEE_Identity_Synchronization_for_Windows`.**
- 3 Type `setup.exe` to execute the installation program.**

The Identity Synchronization for Windows installation wizard is displayed.

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

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

Continue to the next section for Core installation instructions.

On Red Hat Linux

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

▼ To Prepare and Run Identity Synchronization for Windows on Linux

- 1 Log in as root.**
- 2 Change to the directory on the delivery media for Red Hat containing the installation program, `DSEE_Identity_Synchronization_for_Windows`.**

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

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

```
./installer.sh -nodisplay
```

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

Installing Core

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

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

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

Note – You must install the program as root, but after installation you can configure the software to run Solaris and Linux services as a non-root user. (See [Appendix B, “Identity Synchronization for Windows LinkUsers XML Document Sample”](#))

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

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

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

- 1 When the Welcome screen is displayed, read the information provided and then click Next to proceed to the Software License Agreement panel.
- 2 Read the license agreement, then select
 - **Yes (Accept License)** to accept the license terms and go to the next panel.

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

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

Core Install: Configuration Location

Specify information about the configuration directory and root context where the Sun Java(TM) System Identity Synchronization for Windows will be stored or is already stored.

Configuration Directory Host:

Configuration Directory Port: ☐ Secure Port

Configuration Root Suffix:

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.

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

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

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

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

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

Note – Identity Synchronization for Windows encrypts sensitive configuration information before sending it to the configuration Directory Server.

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

Sun Java System Administration Server installed (and configured) as a part of the core components, is installed in a non-SSL mode.

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

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

4 Click Next to open the Configuration Directory Credentials panel.

FIGURE 3–2 Specifying the Administrator Credentials

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

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

Note – If you are not using SSL to communicate with the configuration directory (see [“Installing Core” on page 70](#)), these credentials will be sent without encryption.

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

Core Install: Configuration Password

Please provide a password that will be used to encrypt sensitive parts of the configuration. Remember this password because it must be supplied when you use the console, use the command line utilities, or install other components.

Configuration Password:

Confirm Password:

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 to remember this password as it will be required whenever you want to

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

For information about changing the configuration password see [“Using changepw” on page 208](#).

The Select Java Home panel is displayed (see [“Installing Core” on page 70](#)). The program automatically inserts the location of the Java Virtual Machine directory to be used by the installed components.

Core Install: Select Java Home.

Specify the location of the Java Virtual Machine to be used by installed components.

Java Home Directory:

FIGURE 3-4 Specifying the Java Home Directory

- 8 Verify the Java Home Directory (must be a JDK/JRE 1.5.0_09 or later):**
- If the location is satisfactory, click Next to proceed to the Select Installation Directories panel ([“Installing Core” on page 70](#)).

- If the location is not correct, click Browse to search for and select a directory where Java is installed, for example:
- **On Solaris:** /var/java
- **On Linux:** /usr/bin/java
- **On Windows:** C:\Program Files\jdk1.5

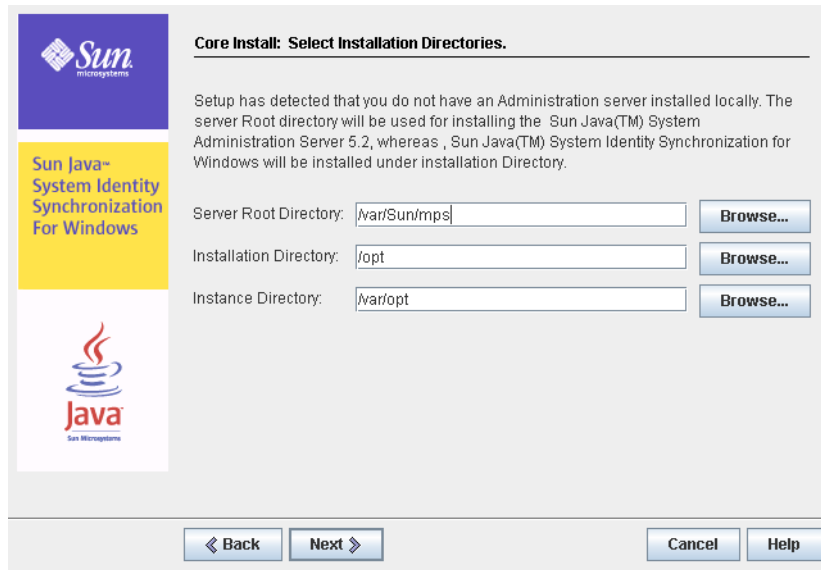


FIGURE 3-5 Specifying the Installation Directories

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 Administration Server installation server root. The Console will be installed in this location.
- **Installation Directory** (*available only when you are installing Core on Solaris or Linux*): Specify the path and directory name of the installation directory. Core binaries, libraries, and executable will be installed in this directory.
- **Instance Directory** (*available only when you are installing Core on Solaris or Linux*): Specify the path and directory name of the instance directory. Configuration information that changes (such as log files) will be stored in this directory.

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

Note – If an Administration Server corresponding to the Configuration Directory Host and Port number provided in step 3 is not found, the installer Administration Server will install the Administration Server as part of the core installation. The default port number for the Administration Server port assigned would be the configuration directory port plus one.

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

Note – You should have installed Message Queue 3.6 Enterprise Edition before starting the Identity Synchronization for Windows installation.

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

On Linux system: Do not install Message Queue and Identity Synchronization for Windows in the same directory.

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

Core Install: Message Queue Configuration

This product requires the use of a pre-existing Message Queue. Specify the installation location, as well as the fully qualified host name and port of the new Broker instance.

Installation Directory:	<input type="text" value="/usr"/>	<input data-bbox="876 1199 948 1222" type="button" value="Browse..."/>
Configuration Directory:	<input type="text" value="/var/lib/mq"/>	<input data-bbox="876 1246 948 1269" type="button" value="Browse..."/>
Fully Qualified Local Host Name:	<input type="text" value="machine1.example.com"/>	
Broker Port Number:	<input type="text" value="7676"/>	

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 (“[Installation Overview](#)” on page 45) is displayed. This panel contains a “To Do” list of the remaining installation and configuration steps. (You also can access this panel from the Console’s Status tab.)

This is a list of remaining installation and configuration steps:

- ✓ 1 : Install the Identity Synchronization core components.
- 2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importcnf'.
- 3 : Prepare every Sun Directory Server included in this configuration by using the console or the 'idsync prepsds' command.
- 4 : Install connectors for every configured directory source.
- 5 : After installing each Sun Directory Server connector, configure the Sun Directory Server plugin on every master and on every read-only replica by using the console or the 'idsync dspluginconfig' command.
- 6 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.
- 7 : Start synchronization using the console or the 'idsync startsync' command.

FIGURE 3-7 To Do List for Identity Synchronization for Windows Installation and Configuration

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

Up to this point, the To Do list will contain a generic list of steps. After you save a configuration, the program provides a list of steps that are customized for your deployment (for example, which connectors you must install).

- 15 After reading the list of steps, click Next and the Start Console Option panel is displayed to indicate you have finished the Core installation.**

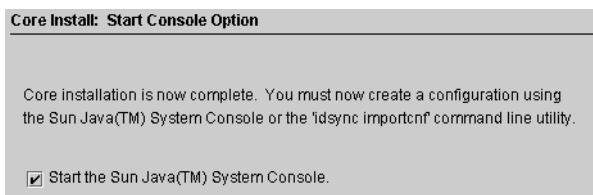


FIGURE 3-8 Starting the Console

- 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 Sun Java System Identity Synchronization for Windows 6.0, you can import an exported version 1.0 or SP1 configuration XML document using the `idsync importcnf` command line utility.

- 17 Click Finished.**
- 18 If you elected to use the Console, the Sun Java System Console Login dialog box is displayed (see “Installing Core” on page 70).**

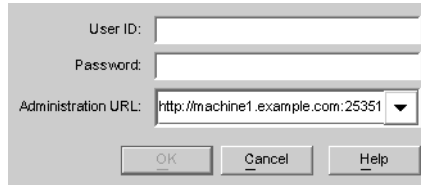


FIGURE 3–9 Logging into the Console

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.

Configuring Core Resources

You must initially configure the Core resources immediately after installing the Identity Synchronization for Windows Core.

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

- “Configuration Overview” on page 79
- “Opening the Identity Synchronization for Windows Console” on page 80
- “Creating Directory Sources” on page 84
- “Selecting and Mapping User Attributes” on page 105
- “Propagating User Attributes Between Systems” on page 110
- “Creating Synchronization User Lists” on page 129
- “Saving a Configuration” on page 133

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

You are not required to configure these resources in a particular order (unless specifically noted in the text); however, using the configuration order presented in this chapter until you become more familiar with the product can save time and prevent errors.

Configuration Overview

This section illustrates the steps you will use to configure the Core resources for your deployment.

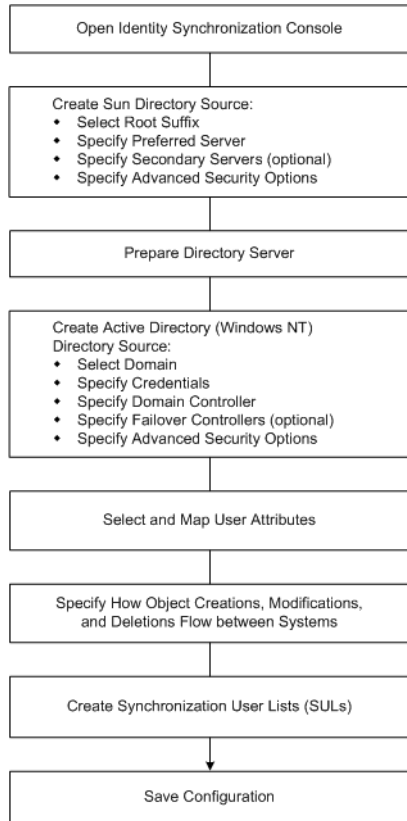


FIGURE 4-1 Configuring Core Resources for Your Deployment

Opening the Identity Synchronization for Windows Console

The Sun Java System Server Console window lists all of the servers and resources under your control and provides information about your system.

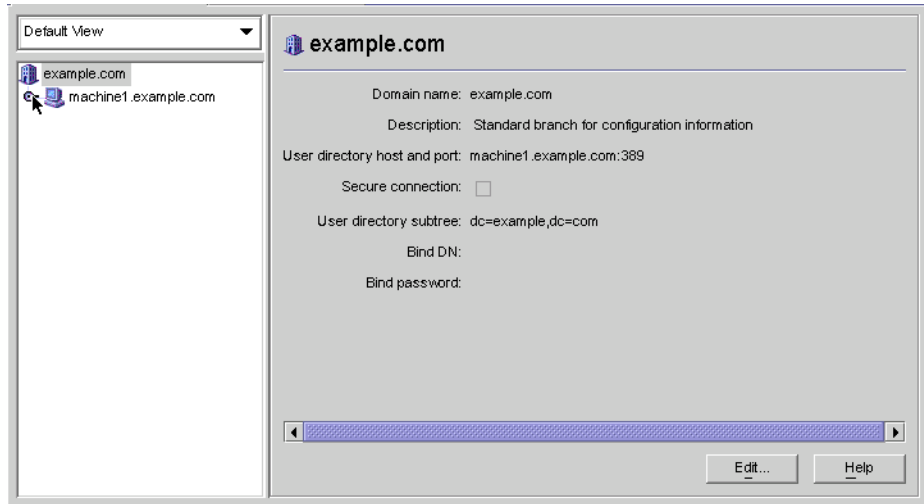


FIGURE 4-2 Sun Java System Server Console

Note – If you have not logged into the Sun Java System Server Console yet, return to [Figure 3-9](#) for instructions.

▼ To Open Identity Synchronization for Windows Console

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

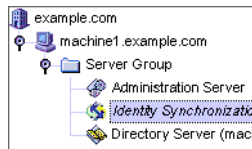


FIGURE 4-3 Expanding the Server Group

The information panel changes to provide information about Identity Synchronization for Windows and your system.

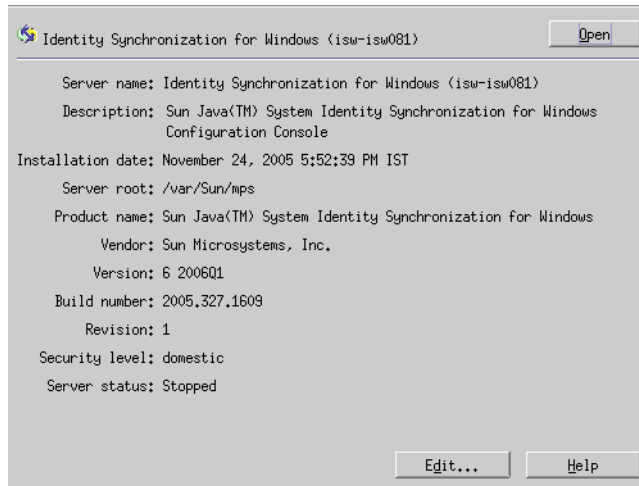


FIGURE 4-4 Information Panel

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

Note – The Edit button (located at the bottom of the panel) enables you to edit the Server name and Description.

- 4 You will be prompted to enter the configuration password that you specified during Core installation. Enter the password and click **OK**.

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

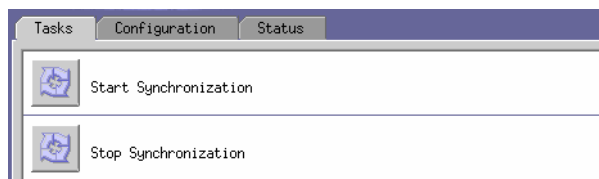


FIGURE 4-5 Console: Tasks Tab

This window contains three tabs:

- **Tasks (Default):** Use this tab to stop and start synchronization between your Sun and Windows systems. (Information about starting and stopping services is provided in [“Starting and Stopping Synchronization” on page 158](#))

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

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

- **Configuration:** Use this tab to configure your systems for synchronization.
- **Status:** Use this tab to do the following:
 - Monitor the status of system components (such as Connectors).
 - View the audit and error logs generated by Identity Synchronization for Windows during configuration and synchronization.
 - Update and check the installation and configuration To Do list.

5 Select the Configuration tab.

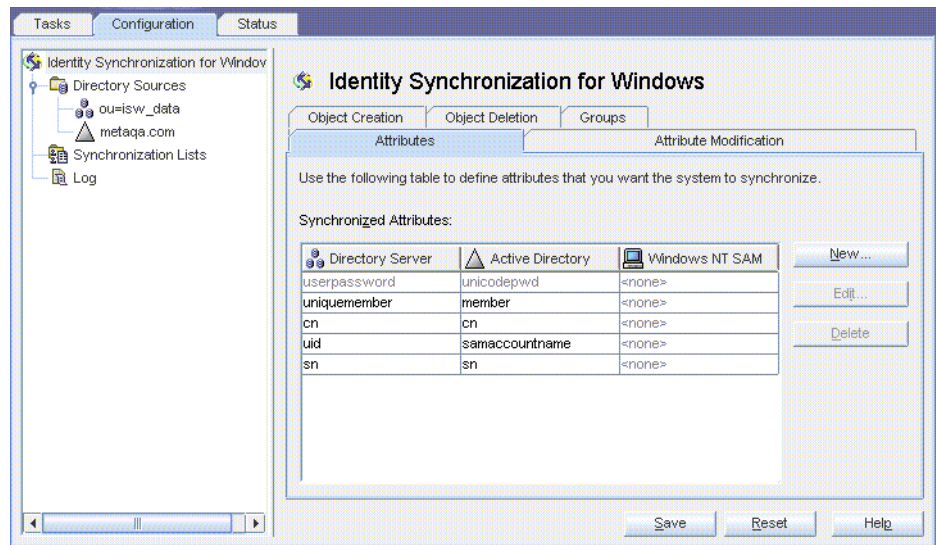


FIGURE 4-6 Console: Configuration Tab

The Configuration panel consists of the following tabs:

- **Attributes:** Use this tab to specify the attributes you want to synchronize between systems.
 - **Attribute Modification:** Use this tab to specify how passwords, attribute modifications, and object disablements are propagated between systems.

- **Object Creation:** Use this tab to specify how newly created passwords and attributes are propagated between systems, and to specify initial values for the objects created by Identity Synchronization for Windows during synchronization.
- **Object Deletion:** Use this tab to specify how deleted passwords and attributes are propagated between systems.

You must configure at least one Sun Java System Directory Server directory source, and at least one Windows server directory source (either Active Directory or Windows NT). Proceed to the next section for instructions.

Creating Directory Sources

▼ To Create Directory Sources

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

- 1 ["Creating a Sun Java System Directory Source" on page 85](#)
- 2 ["Preparing Sun Directory Source" on page 91](#)
- 3 ["Creating an Active Directory Source" on page 95](#)
- 4 ["Creating a Windows NT SAM Directory Source" on page 102](#)

Note – At minimum, you must configure at least one Sun Java System Directory source and at least one Windows directory source (Active Directory and/or NT SAM).

Select the Directory Sources node in the navigation tree and the Directory Sources panel is displayed.

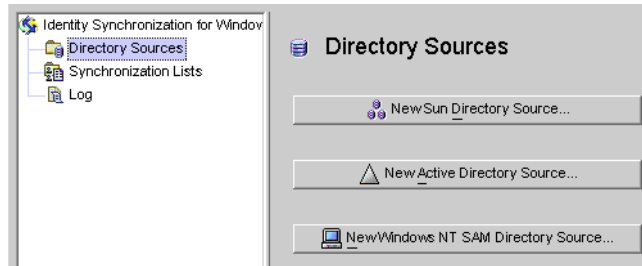


FIGURE 4-7 Accessing the Directory Sources Panel

Creating a Sun Java System Directory Source

Each Sun Java System directory source is associated with a Connector and set of Plug-ins that can be deployed in a replication scenario involving multiple servers. The Directory Server Connector is capable of synchronizing changes from Windows directory source to the preferred server (master). In case, the preferred server is down, the changes will failover to the secondary server in the configured secondary servers list in a sequential manner till the preferred server comes up. Directory Server replication will replicate changes made from the preferred server (master) to other preferred secondary servers configured in the topology. Any Directory Server Plug-in can handle password validity checks from Windows directory sources and users can change passwords at any server.

▼ To Create a New Sun Java System Directory Source

- 1 Click the New Sun Directory Source button to invoke the Define Sun Java System Directory Source wizard.

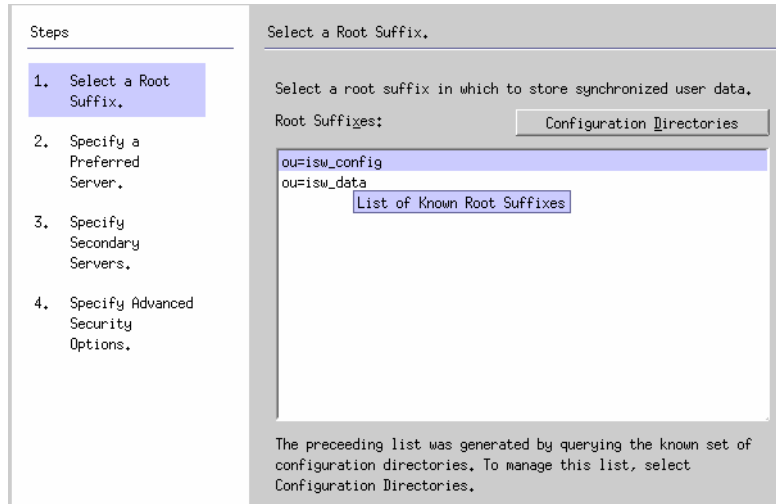


FIGURE 4-8 Selecting a Root Suffix

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.**

If the root suffix you want to synchronize with is not affiliated with a configuration directory registered with Identity Synchronization for Windows, then you must specify a new configuration directory, as follows:

- a. Click the Configuration Directories button to specify a new configuration directory.
- b. When the Configuration Directories dialog box is displayed ([Step 3](#)), click the New button to open the New Configuration Directories dialog box.

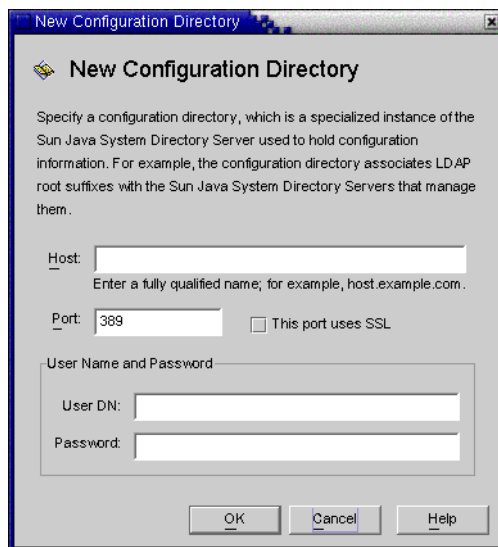


FIGURE 4-9 Selecting a New Configuration Directory

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.

Editing and Removing Configuration Directories

You can also use the Configuration Directories dialog box to manage your list of configuration directories, as follows:

- Select a configuration directory from the list pane, and then click the Edit button. When the Edit Configuration Directories dialog is displayed, you can change the Host, Port, Secure Port, User Name, and Password parameters.
 - Select a configuration directory from the list pane, and then click Remove to delete the directory from the list.
- d. **Click OK to close the Configuration Directories dialog box and the newly selected configuration directory's root suffixes are displayed in the list pane.**
- By default, Directory Server creates a root suffix whose prefix corresponds to the components of the machine's DNS domain entry. It uses the following suffix:
- `dc=your_machine's_DNS_domain_name`
- That is, if your machine domain is *example.com*, then you should configure the suffix `dc=example`, `dc=com` for your server. The entry named by the chosen suffix must already exist in the directory.
- e. **Select the root suffix, and click Next.**

The Specify Preferred Servers panel is displayed (see “[Creating a Sun Java System Directory Source](#)” on page 85).

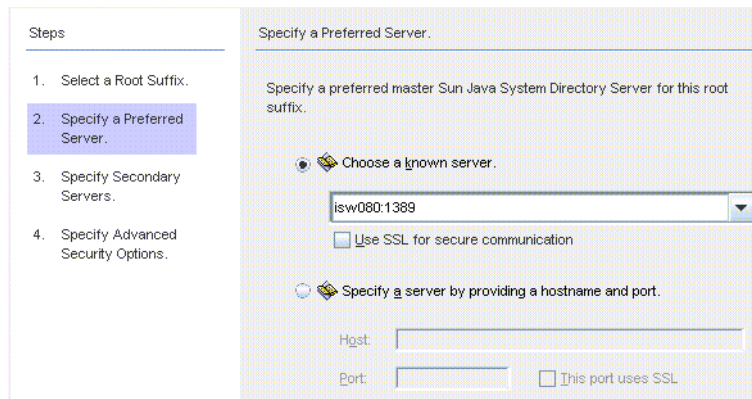


FIGURE 4-10 Specifying a Preferred Server

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 Server system.

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

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

- Select the Choose a Known Server option, and then select a server name from the drop-down list.

Note – The Directory Server must be running to appear in the list. If the server is down temporarily, select the Specify a Server by Providing a Hostname and Port option, and then enter the server information manually.

Enable the Use SSL for secure communication box if you want the Directory Server to communicate using SSL. However, if you enable this feature there are some additional setup steps you must perform after installation. For more information, see [“Enabling SSL in Directory Server” on page 181](#)

- Select the Specify a Server By Providing a Hostname and Port option, and then type the server’s Host name and Port into the text fields.

Select the This Port Uses SSL checkbox if the port you specified uses SSL.

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

FIGURE 4-11 Specifying the Secondary Servers for Failover Support

You can add, edit, or delete the Secondary Servers:

- Click the **New** button to display the **Add Sun Directory Source** dialog box. Enter the host name, port, user DN, password, and then click **OK**. For more information on these fields, see [Step c](#).
 - Click the **Edit** button to display the **Edit Sun Directory Source** dialog box. Enter the host name, port, user DN, password, and then click **OK**. For more information on these fields, see [Step c](#).
 - From the **Secondary Servers** list, select the server you want to delete and click the **Remove** button.
- 5 To specify the secondary Directory Servers, select a server name from list, and then click **Next**.

Note –

- The Directory Server must be running or the server name will *not* appear in list.
- Do not use the same host name and port for both the preferred and the secondary servers in a Sun directory source.
- If you enable the Secure Port feature, there are additional setup steps you must perform after installation. For more information, see [“Enabling SSL in Directory Server” on page 181](#)

If you do not want to specify a secondary server, click **Next**.

- 6 If you want to use secure SSL communication, read the notes below, and then enable one or both of the following options:

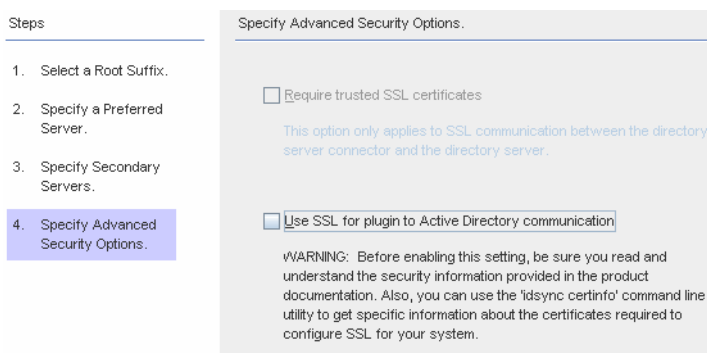


FIGURE 4-12 Specifying Advanced Security Options

Note – You must install the Directory Server Plug-in on each Directory Server (any master, replica, or hub) where users will bind or where passwords will be changed.

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

- To encrypt channel communication only or to encrypt channel communication and use certificates to ensure participants' identity verification between Directory Server and the Directory Server Connector, enable the Require Certificates for SSL box.

Clear the checkbox if you do not want to trust certificates.

- To use secure SSL communication between the Directory Server Plug-in and Active Directory, enable the Use SSL for Plug-in to Active Directory communication box.

If you enable these features, then additional setup is required after installation. See [“Enabling SSL in Directory Server” on page 181](#)

- You can use the `idsync certinfo` command line utility to determine which certificates you must add for each Directory Server Plug-in and/or Connector certificate database. See [“Using certinfo” on page 208](#)
- If your primary and secondary Directory Servers are part of a multimaster replication (MMR) deployment, refer to [Appendix E, “Identity Synchronization for Windows Installation Notes for Replicated Environments”](#)

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

The program adds the selected directory sources to the navigation tree under Directory Sources, and the Prepare Directory Server Now? dialog is displayed.

You must prepare the Directory Server to be used by Identity Synchronization for Windows. You can choose to perform this task now, or you can do it later — but you must prepare the Directory Server before you install the Connectors. (Instructions for installing Connectors are provided in [Chapter 5, “Installing Connectors”](#)).

- If you want to prepare the Directory Server now, click Yes to open the wizard, and then proceed to the next section, [“Preparing Sun Directory Source” on page 91](#)
- If you prefer to perform this process later, click No and proceed to [“Creating an Active Directory Source” on page 95](#).

Preparing Sun Directory Source

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

Preparing the Directory Server:

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

Note –

- As an alternative to using the Console, you can use the `idsync preps` command line utility to prepare the Directory Server. For more information, see [“Using preps” on page 210](#).
- To prepare the Directory Server using the `idsync preps` command line utility, you must know which hosts and suffixes you will be using and you must have Directory Manager’s credentials.

You can use the Prepare Directory Server wizard to prepare the Directory Server.

FIGURE 4–13 Entering Your Directory Manager Credentials

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

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

▼ To Prepare your Directory Server Source

1 Enter the following credentials for the Directory Manager account.

- **Directory Manager User Name**
- **Directory Manager Password**

If you are using a secondary host (MMR configurations), then the Secondary Host options will be active and you must specify credentials for these hosts too.

2 When you are done, click Next and the Specify Preparation Configuration panel is displayed.

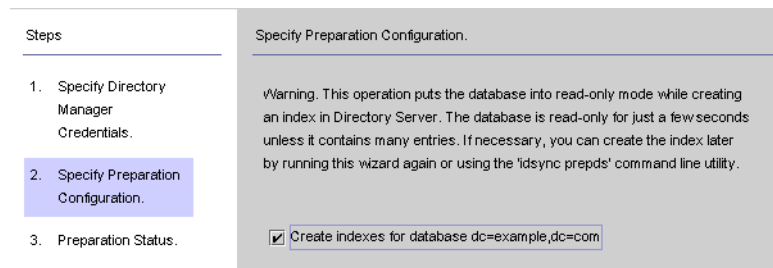


FIGURE 4-14 Specifying the Preparation Configuration

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

Note –

- This operation can take some time, depending on the size of your database.
- While your database is in read-only mode, any attempts to update information in the database will fail.
- Taking your database offline enables you to create the indexes much faster.
- To create the indexes now, enable the Create indexes for database box, and then click Next.
- To create the indexes later (either manually or by running this wizard again) clear the Create indexes for database box, and then click Next.

3 The Preparation Status panel is displayed to provide information about the Directory Server preparation progress.

- When a SUCCESS message is displayed at the bottom of the message pane, click Finish.
- If error messages display, you must correct the problem(s) reported before you can continue. Check the error logs (see the Status tab) for more information.

- 4 Return to the Configuration tab in the Console. Select the Sun Directory source node in the navigation tree to view the Sun Directory Source panel.

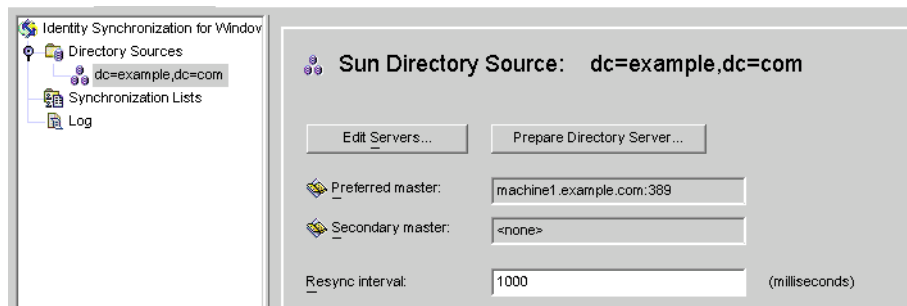


FIGURE 4-15 Sun Directory Source Panel

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” on page 85](#).

Note – If you recreate 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 prepd`s or click the Prepare Directory Server button after selecting the appropriate Sun directory source in the Console.

- **Prepare Directory Server:** Click this button and follow the instructions for [“Preparing Sun Directory Source” on page 91](#) to prepare a Directory Server.

If anything changes on the Directory Server after you initially prepare the server (for example, if an index is deleted or you lose the Retro-Changelog database), you can re-prepare the server.

- **Resync interval:** Specify how often you want the Directory Server Connector to check for changes. (*Default is 1000 milliseconds*)

- 5 Add a Directory Server directory source for each user population in your Sun Java System Directory Server enterprise that you want to synchronize.

When you are finished, you must create at least one Windows directory source:

- To create an Active Directory source, continue to the next section, “[Creating an Active Directory Source](#)” on page 95.
- To create a Windows NT directory source, continue to “[Creating a Windows NT SAM Directory Source](#)” on page 102

Creating an Active Directory Source

You should add an Active Directory directory source for each Windows domain in your network that you want to synchronize.

Each Active Directory deployment has at least one global catalog that knows about all the global information across all Active Directory domains. To access the global catalog, the rights assigned to a normal user are sufficient unless you change the default permissions.

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

▼ To Configure and Create Windows Active Directory Servers in a Network

Perform these steps if there are Windows Active Directory servers in your network:

- 1 **Select the Directory Sources node in the navigation tree, and then click the New Active Directory Source button on the Directory Sources panel.**

The Windows Global Catalog dialog box is displayed.

Windows Global Catalog

The system requires a global catalog from which to discover schema and topology information about the Windows domain. Specify the host and access credentials for the global catalog in your Windows domain.

Host:
Enter a fully qualified name; for example, host.example.com.

☐ This port uses SSL

Directory Source Credentials:

User DN:

Password:

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.

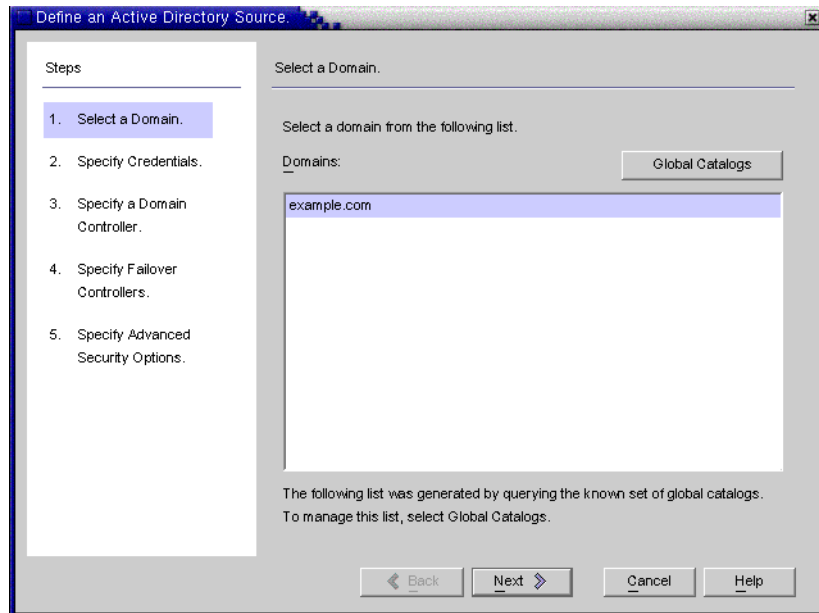


FIGURE 4-17 Define an Active Directory Source Wizard

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 and click OK.

If the domain you want to use is not displayed in the list, you must add the global catalog that knows about that domain using the following steps:

a. Click the Global Catalogs button and the Global Catalogs wizard is displayed.

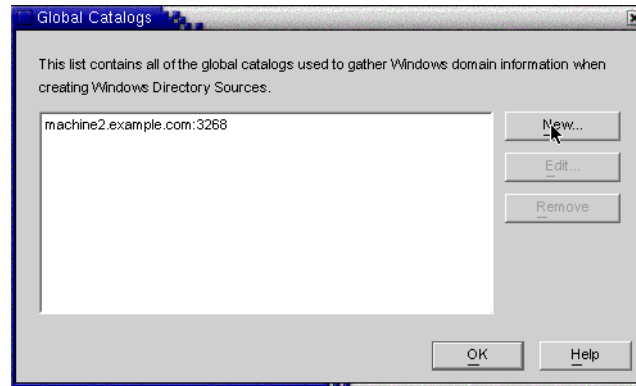


FIGURE 4-18 Specifying a New Global Catalog

- 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 in Step 2), and then click OK**
 - d. The new global catalog and port, are displayed in the Global Catalogs list panel. Select the catalog name, and then click OK.**
 - e. Repeat these steps if you want to add more global catalogs (domains) to the system.**
 - f. When you are done, click the Next button in the Select a Domain pane.**
- 5 When the Specify Credentials panel is displayed, review the value in the User DN field.**

Steps

1. Select a Domain.
2. Specify Credentials.
3. Specify a Domain Controller.
4. Specify Failover Controllers.
5. Specify Advanced Security Options.

Specify Credentials.

Specify credentials for accessing user entries in all servers in this directory source.

User DN:

Password:

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 rights will depend on the direction of synchronization, as follows:

- If you are configuring synchronization flow from Active Directory to Directory Server only, then the user provided for the Active Directory Connector does not require many special privileges. A normal user with the extra privilege to “Read All Properties” in the domain being synchronized will suffice.
 - If you are configuring synchronization flow from Directory Server to Active Directory, then the Connector user must have more privileges because, synchronization changes the user entries in Active Directory. In this setup, the Connector user must have either the “Full Control” privilege or be a member of the Administrators group.
-

6 Click Next to open the Specify a Domain Controller panel.

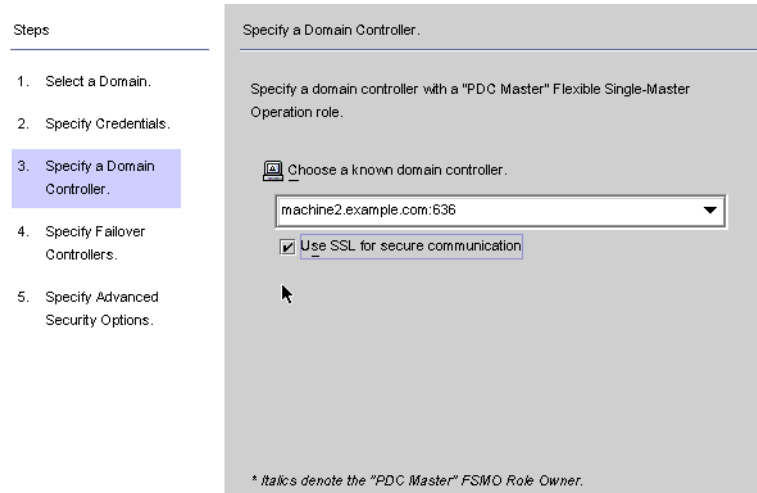


FIGURE 4–20 Specifying a Domain Controller

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 flexible single master operation (FSMO) role for synchronization.

By default, password changes made at all domain controllers will be replicated immediately to the Primary Domain Controller FSMO role owner, and if you select this domain controller, Identity Synchronization for Windows will synchronize these password changes immediately to the Directory Server.

In some deployments, the `AvoidPdcOnWan` attribute may be set in the Windows registry because there is a significant network “distance” to the PDC, which will delay synchronization significantly. (See *Microsoft Knowledge Base Article 232690* for more information.)

- 7 **Select a domain controller from the drop-down list.**
- 8 **If you want the Identity Synchronization for Windows Connector to communicate with the domain controller over a secure port, enable the Use a Secure Port box.**

Note – The program automatically installs the CA certificate in the Active Directory Connector if you are using Microsoft certificate server. If you are not, then you must manually add the CA certificate in the Active Directory Connector (see [“Enabling SSL in the Active Directory Connector” on page 183](#) change your flow settings after initial configuration these procedures apply as well).

9 When you are done, click Next.

The Specify Failover Controllers panel is displayed (see [“Creating an Active Directory Source” on page 95](#)). You can use this panel to specify any number of failover domain controllers.

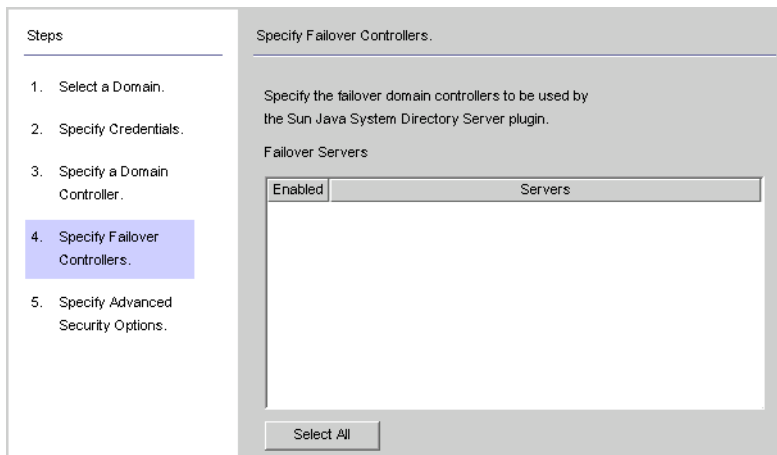


FIGURE 4-21 Specifying Failover Controllers

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 Plug-in will communicate with any number of domain controllers when validating password changes to Directory Server.

If Directory Server tries connecting to an Active Directory domain controller and that domain controller is not available, Directory Server will iteratively try connecting to the failover domain controller(s) specified.

- 10 **Select one or more of the server names listed in the Failover Servers list pane (or click the Select All button to specify all of the servers in the list), and then click Next.**
- 11 **The Specify Advanced Security Options panel is displayed.**

The Require trusted SSL certificates option is active (available for selection) only if you enabled the Use SSL for Secure Communication box on the Specify a Domain Controller panel.

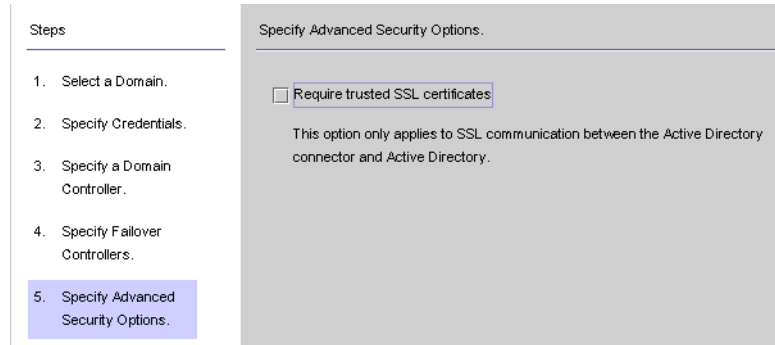


FIGURE 4–22 Specifying Advanced Security Options

- 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 setup 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 185.

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

The program adds the newly specified Active Directory source to the navigation tree under Directory Sources.

13 Select the Active Directory source node to view the Active Directory Source panel.

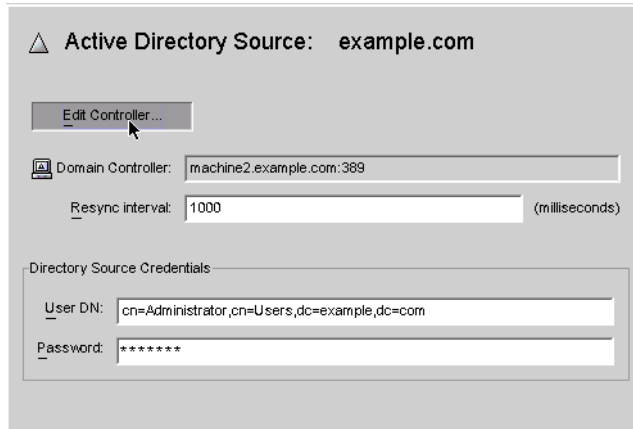


FIGURE 4-23 Active Directory Source Panel

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” on page 95](#).
- **Resync Interval:** Specify how often you want the Active Directory Connector to check for changes. (*Default is 1000 milliseconds*)
- **Directory Source Credentials:** Change the specified User DN and/or password.

Creating a Windows NT SAM Directory Source

This section explains how to create a Windows NT SAM Directory Source where you can deploy Identity Synchronization for Windows.

▼ To Deploy Identity Synchronization for Windows on Windows NT

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



FIGURE 4-24 Directory Sources Panel

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

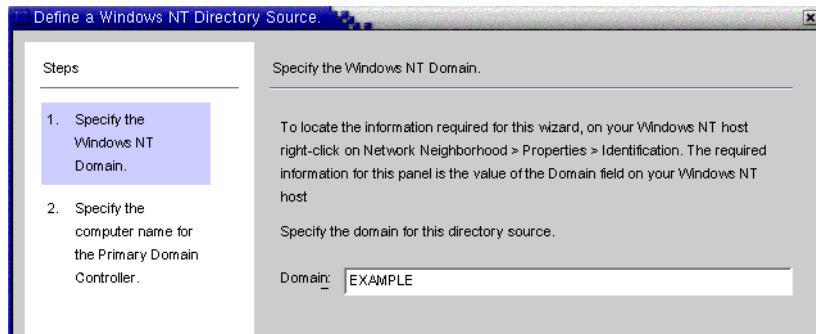


FIGURE 4-25 Specifying a Windows NT SAM Domain Name

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

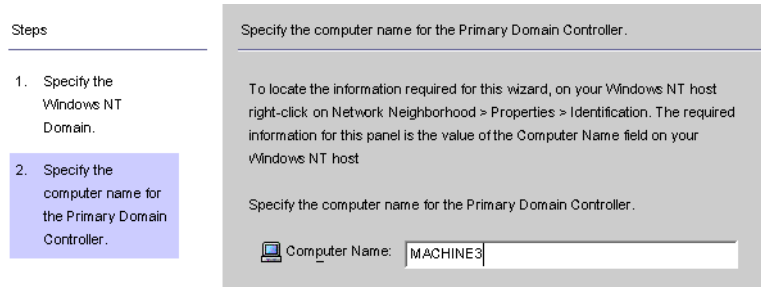


FIGURE 4-26 Specifying a Name for the Primary Domain Controller

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.

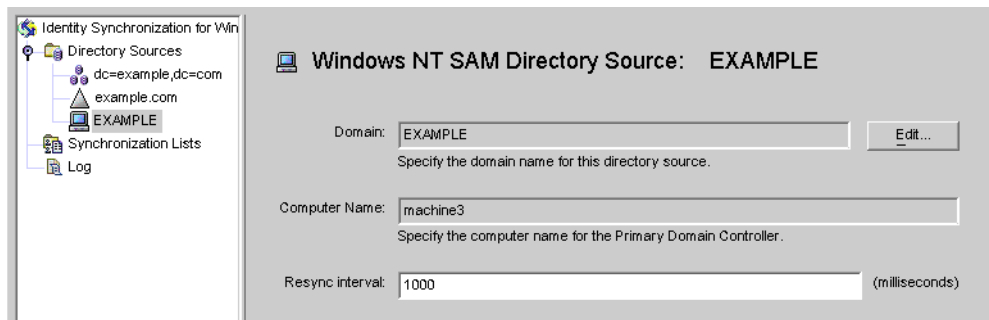


FIGURE 4-27 Windows NT SAM Directory Source Panel

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” on page 95](#).
- **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 105](#)

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” on page 105](#)
- [“Creating Parameterized Default Attribute Values” on page 107](#)
- [“Changing the Schema Source” on page 108](#)

Selecting and Mapping Attributes

There are two types of attributes:

- **Significant:** Attributes that are synchronized between systems when you create or modify user entries.
- **Creation:** Attributes that are synchronized between systems only when you create user entries.

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

This section explains how to select user attributes for synchronization and how to map these attributes (one-to-one) so that when you specify an attribute for Directory Server the equivalent attribute will display in your Active Directory and/or Windows NT environment (and vice versa), and the companion Windows attributes will have their values synchronized.

▼ To Select and Map Attributes for Synchronization

- 1 Select the Identity Synchronization for Windows node at the top of the navigation tree.

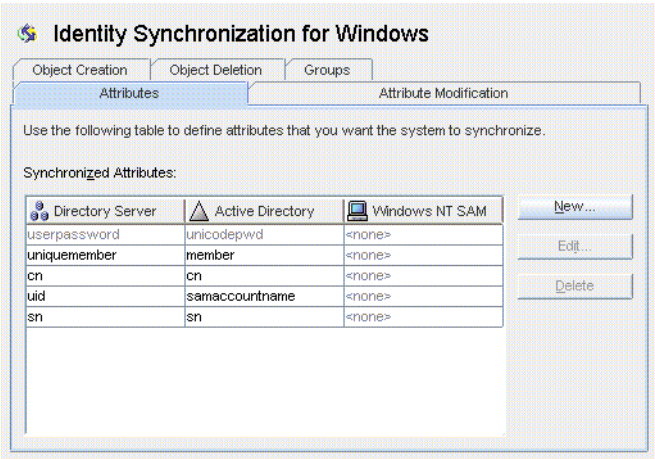


FIGURE 4-28 Attributes Tab

Note – When the Group Synchronization feature has been enabled, the *uniquemember* (Directory Server) attribute and *member attribute* (Active Directory) are internally mapped and would be indicated as shown in the console.

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

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

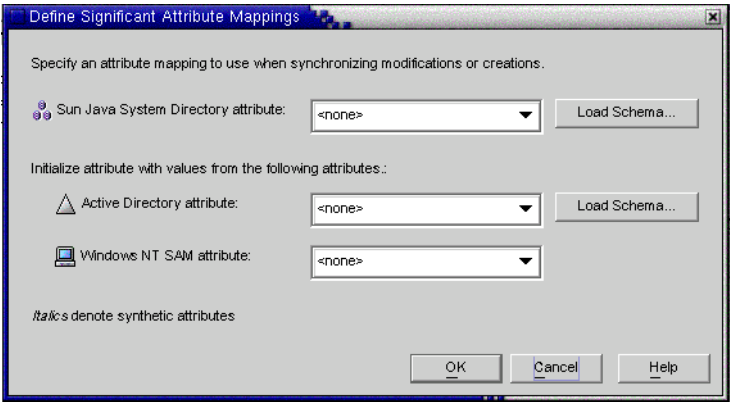


FIGURE 4-29 Defining Significant Attribute Mappings

Note – Which creation attributes are mandatory for Directory Server (or for Active Directory) will depend on the objectclass configured for your Sun-side (or Active Directory-side) user entries.

The program automatically uses *inetOrgPerson* as the default objectclass for Directory Server, and you loaded the Active Directory schema when you specified the global catalog. So you do not use the Load Schema buttons unless you want to change the default schema.

If you want to change the default schema source, see [“Changing the Schema Source” on page 108](#)

- 3 **Select an attribute from the Sun Java System attribute drop-down list (for example *cn*), and then select the equivalent attribute from the Active Directory attribute and/or Windows NT SAM attribute drop-down menus.**
- 4 **When you are finished, click OK.**
- 5 **To designate additional attributes, repeat steps 2 through step 4.**

A finished Synchronized Attributes table might look something like the following example, which shows the userpassword, cn, and telephonenumber Directory Server attributes mapped to unicodepwd, cn, and telephonenumber Active Directory attributes.

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

FIGURE 4-30 Completed Synchronized Attributes Table

Creating Parameterized Default Attribute Values

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

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

When you create these attribute values:

- You can use multiple attributes in a creation expression (cn=%givenName% %sn%).
- If A=0, then B can have one default value only.
- You can use the backslash symbol (\\) for quoting (for example, diskUsage=0\\%).
- Do not use expressions that have cyclic substitution conditions (for example, if you specify description=%uid%, you cannot use uid=%description%.)

Note – When Group Synchronization is enabled, the following are important:

1. The creation expression supported at Active Directory is `cn=%cn%`.
2. The creation expression must contain valid attribute names belonging to the group objectclass also since the creation expression is common to both user as well as the group.
For example: The attribute `sn` is not part of the `groupofuniqueNames` objectclass at the Directory Server. Hence the following creation expression would be invalid for a group object. (Though it would work fine for user.)
`cn=%cn%. %sn%`
3. The attribute used in the creation expression must be provided with a value for every user/group entry created. The value maybe provided using the command line interface, if the console does not have the provision.

Changing the Schema Source

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

▼ To Change the Default Schema Source

- 1 **Click the Load Schema button on the Define Significant Attribute Mappings dialog box.**
The Select Schema Sources panel is displayed.

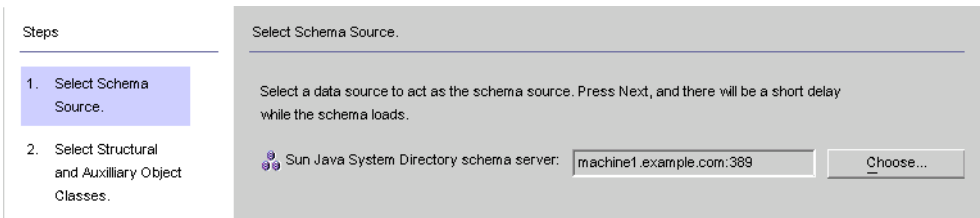


FIGURE 4-31 Selecting Schema Sources

Use this panel to specify from which Sun Java System Directory Server schema server you want to read the schema. This schema contains the object classes that are available on your system, and object classes define which attributes are available for users on your system.

The program adds your configuration directory to the Sun Java System Directory schema server field by default.

2 To select a different server, click the Choose button.

The Select a Sun Schema Host dialog box is displayed. This dialog box contains a list of the configuration directories that gather administrative information about your directory sources.

From this dialog box, you can:

- Create new configuration directories and add them to the list.
Click New, and when the New Configuration Directory dialog box displays; specify a Host, Port, User DN, and Password. Click OK when you are done.
- Edit existing directories.
Click Edit, and when the Edit Configuration Directory dialog box displays, you can change the Host, Port, User DN, and/or Password. Click OK when you are done.
- Remove directories from the list.
Select a directory name from the list and then click the Remove button.

3 Select a server from the list and click OK when you are done. (Generally, one of your Sun synchronization host(s) is a good choice as a schema source.)

4 Click the Next button and the Select Structural and Auxiliary Object Classes panel is displayed.

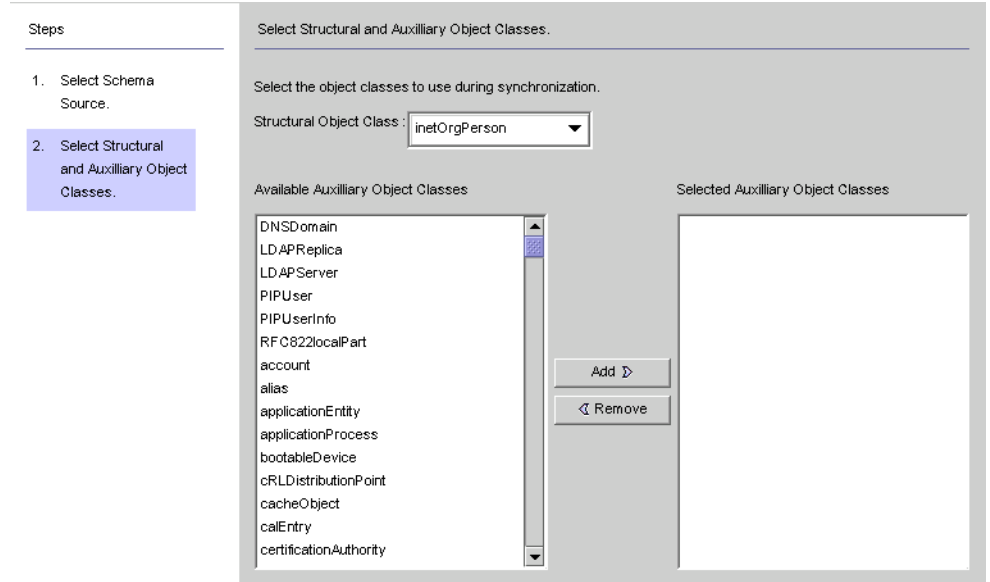


FIGURE 4-32 Selecting Structural and Auxiliary Object Classes

Use this panel to specify the object classes to synchronize, as follows:

- **Structural Object Class:** Every entry that is created or synchronized from the selected Directory Server must have at least one structural object class.
- **Auxiliary Object Classes:** These object classes augment the selected structural class and provide additional attributes for synchronization.
To specify structural and auxiliary object classes:
 - a. **Select a structural object class from the drop-down list.** (*Default is inetorgperson.*)
 - b. **Select one or more object classes from the Available Auxiliary Object Classes list pane, and then click Add to move your selection(s) to the Selected Auxiliary Object Classes list pane.**
The selected object class(es) determine which Directory Server source attributes will be available for selection as significant or creation attributes. The object class(es) also determine the mandatory creation attributes.

To delete selections from the Selected Auxiliary Object Classes list, click the object class name and then click the Remove button.
 - c. **When you are done, click Finish and the program loads the schema and selected object classes.**

Propagating User Attributes Between Systems

After you create and map the user attributes you want to synchronize, you must tell Identity Synchronization for Windows how to propagate (flow) the attribute creations, modifications, and deletions between your Directory Server and Windows Systems.

By default, Identity Synchronization for Windows:

- Synchronizes from Windows to Directory Server only
- Synchronizes the password attribute only (unless you specified significant attributes in the previous section)
- Does not synchronize the creation or deletion of entries

This section explains how to configure attribute synchronization between systems. The information is organized as follows:

- [“Specifying How Object Creations Flow” on page 111](#)
- [“Specifying How Object Modifications Flow” on page 116](#)
- [“Specifying Configuration Settings for Group Synchronization” on page 124](#)
- [“Configuring and Synchronizing Account Lockout and Unlockout” on page 125](#)
- [“Specifying How Deletions Flow” on page 128](#)

Specifying How Object Creations Flow

▼ To Specify How Object Creations Should Flow Between Directory Server and Active Directory Systems

- 1 Click the Object Creation tab.

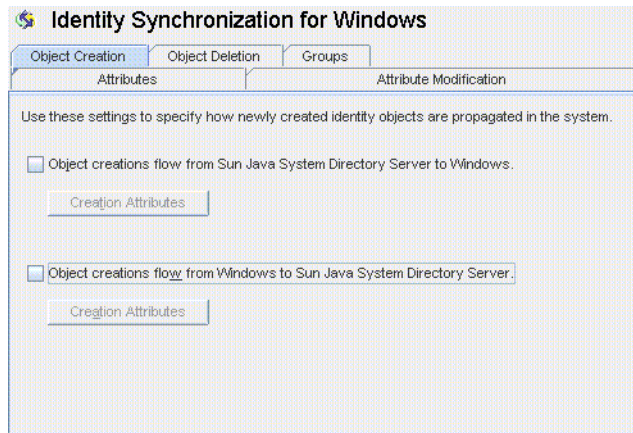


FIGURE 4-33 Selecting and Propagating Creations

- 2 You can enable or disable the flow of creations as follows:
 - Enable **Object creations flow from Sun Java System Directory Server to Windows** to propagate creations from the Directory Server environment to your Windows servers.
 - Enable **Object creations flow from Windows to Sun Java System Directory Server** to propagate creations from the Windows environment to your Directory Servers.
 - Enable both options for bidirectional flow.
 - Disable both options to prevent user creations from propagating from one system to the other. (*Default*).
- 3 To add, edit, or delete creation attributes to synchronize between systems, click the **Creation Attributes** button located under the selected option(s).

The Creation Attribute Mappings and Values dialog box displays.

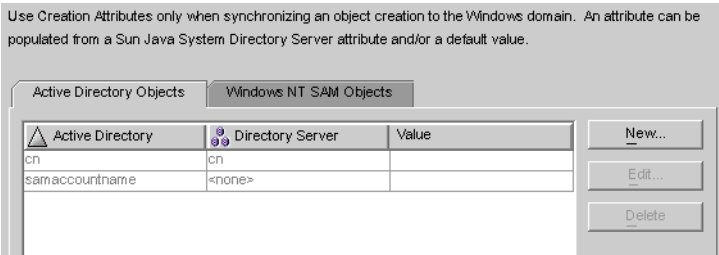


FIGURE 4-34 Creation Attributes Mappings and Values: Directory Server to Windows

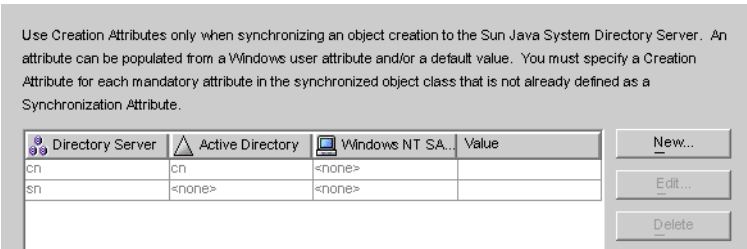


FIGURE 4-35 Creation Attributes Mappings and Values: Windows to Directory Server

You can use either of the dialog boxes to specify new creation attributes, edit, or delete existing attributes. For more information, see [“Specifying New Creation Attributes” on page 112](#).

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 105](#)).

Specifying New Creation Attributes

The following instructions explain how to add and map creation attributes from Active Directory to Directory Server. (The procedure for adding and mapping creation attributes flowing from Directory Server to Windows and from Windows to Directory Server is similar.)

▼ To Specify New Creation Attributes

- 1 Click the New button in the Creation Attribute Mappings and Values dialog box.

The Define Creation Attribute Mappings and Values dialog box is displayed.

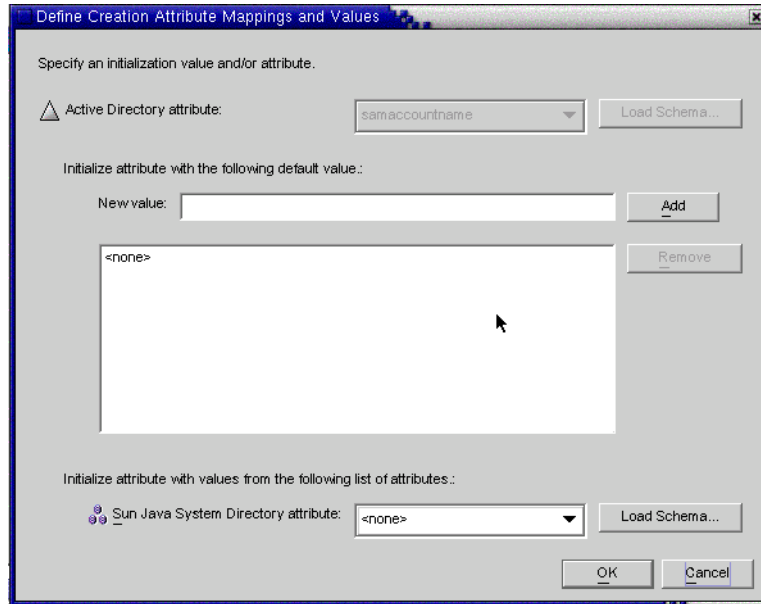


FIGURE 4-36 Defining Creation Attribute Mappings and Values

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



FIGURE 4-37 Selecting a New Active Directory Attribute

Identity Synchronization for Windows allows you to initialize an attribute with multiple values—if the attribute itself accepts multiple values.

For example, if your company has three fax telephone numbers, you can specify the `facsimiletelephonenumber` 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.

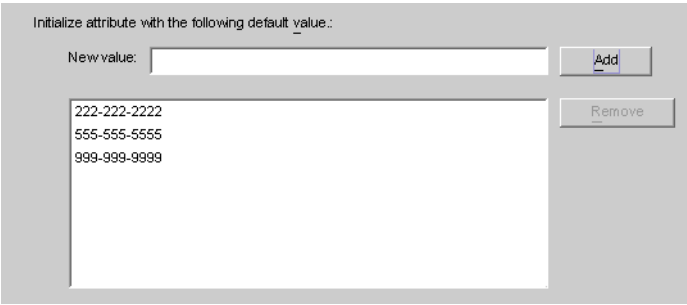


FIGURE 4-38 Specifying Multiple Values for a Creation Attribute

- 4 To map the attribute to Directory Server, select an attribute name from the Directory Server attribute drop-down list.

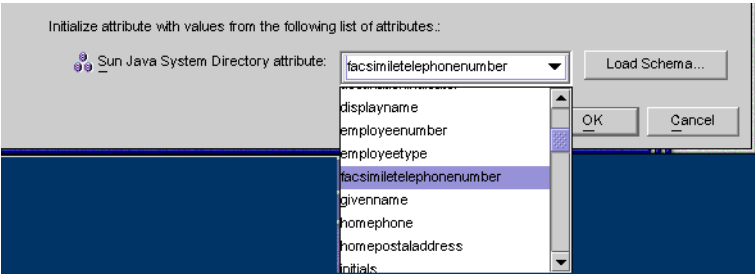


FIGURE 4-39 Mapping the Directory Server Attribute

- 5 When you are finished, click OK.

Based on the example, the finished Creation Attributes and Mappings table would look like the one in the following figure.

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

FIGURE 4-40 Completed Creation Attributes and Mappings Table

- 6 To designate additional attributes, repeat these steps.

Editing Existing Attributes

▼ To Edit Creation Attributes Mapping or Values

- 1 Select the **Object Creation** tab, and click on the **Creation Attributes** button located under the selected creation option.
- 2 When the **Creation Mappings and Values** dialog box is displayed, select the attribute from the table, and then click the **Edit** button.
The **Define Creation Mappings and Values** dialog box is displayed.
- 3 Use the drop-down menus to change the existing mapping between **Directory Server** and **Active Directory (or Windows NT)**.
For example, if you have Sun Java System Directory Server's homephone attribute mapped to Active Directory's other telephone attribute. You could use the Active Directory attributes drop-down list to change the mapping to homephone.
- 4 You can also add or remove attribute values:
 - To add a value, enter the information in the **New Value** field and click **Add**.
 - To remove a value, select the value from the list pane and click **Remove**.
- 5 When you are done, click **OK** to apply your changes and close the **Define Creation Mappings and Values** dialog box.
- 6 Click **OK** again to close the **Creation Mappings and Attributes** dialog box.

Removing Attributes

▼ To Remove Creation Attributes Mapping or Values

- 1 Select the **Object Creation** tab, and click the **Creation Attributes** button located under the selected creation option.
- 2 When the **Creation Mappings and Values** dialog box is displayed, select the attribute from the table, and then click the **Delete** button.
The attribute is removed from the table immediately.
- 3 When you are done, click **OK** to close the **Creation Mappings and Attributes** dialog box.

Specifying How Object Modifications Flow

Use the Attribute Modification tab to control how modifications made to user attributes and passwords will be propagated (flow) between your Sun and Windows systems.

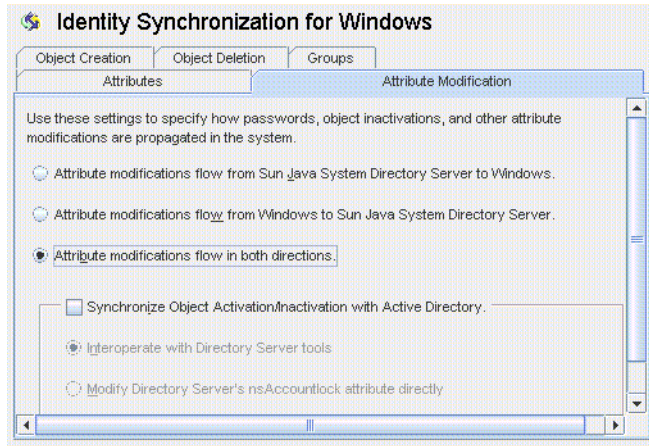


FIGURE 4-41 Attribute Modification Tab

You use this tab to configure the following:

- Specify the direction in which modifications flow between Directory Server and Windows directory sources.
- Control whether object activations and inactivations (*enables* and *disables* on Active Directory) will be synchronized between Directory Server and Active Directory sources, and specify the method in which user accounts are activated and inactivated.

Note – You cannot synchronize account statuses with Windows NT directory sources.

Specifying Direction

Select one of the following buttons to control how changes made in the Directory Server and Windows environments will be propagated between systems.

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

- **Attribute modifications flow in both directions** : Propagates changes bidirectionally (from one environment to the other environment).

Configuring and Synchronizing Object Activations and Inactivations

If you enable the Synchronize Object Activations/Inactivations with Active Directory box you can synchronize object activations and inactivations (known as *enables* and *disables* on Active Directory) between Directory Server and Active Directory sources.

Note – You cannot synchronize activations and inactivations with Windows NT directory sources.

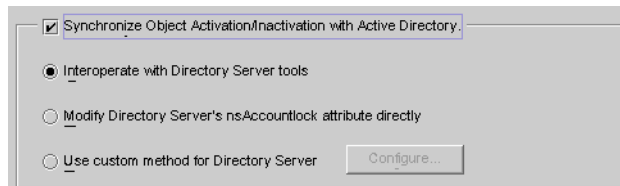


FIGURE 4–42 Synchronizing Object Activations and Inactivations

▼ 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:
 - [“Interoperating with Directory Server Tools” on page 117](#)
 - [“Modifying Directory Server’s NsAccountLock Attribute Directly” on page 118](#)

Note – These options are mutually exclusive.

- [“Using a Custom Method for Directory Server” on page 119](#)

Interoperating with Directory Server Tools

Select this option if you use the Directory Server Console or command line tools to activate/inactivate an object. With this option selected Identity Synchronization for Windows cannot set or remove the nsAccountLock attribute directly. In addition, the program cannot detect objects that have been inactivated using other roles such as cn=nsdisabledrole,

database suffix or roles that nest within other roles, such as *cn=nsdisabledrole, database suffix* or *cn=nsmanageddisabledrole, database suffix*.

- To activate objects, Identity Synchronization for Windows will remove the *cn=nsmanageddisabledrole, database suffix* value from the *nsroledn* attribute.
- To inactivate objects, Identity Synchronization for Windows will add the *cn=nsmanageddisabledrole, database suffix* value to the *nsroledn* attribute.

Note – If you enable the Interoperate with Directory Server Tools option, Identity Synchronization for Windows cannot set or remove the *nsAccountLock* attribute directly. In addition, Identity Synchronization for Windows cannot detect objects have been inactivated using other roles.

For example, *cn=nsdisabledrole, database suffix* or roles that nest within other roles such as *cn=nsdisabledrole, database suffix* or *cn=nsmanageddisabledrole, database suffix*.

Interoperating with Directory Server Tools describes how Identity Synchronization for Windows detects and synchronizes object activations/inactivations when you enable the Interoperate with Directory Server Tools option.

TABLE 4-1 Interoperating with Directory Server Tools

Activations	Inactivations
Identity Synchronization for Windows detects an activation only when the <i>cn=nsmanageddisabledrole, database suffix</i> role is removed from the object.	Identity Synchronization for Windows detects an inactivation only when the entry's <i>nsroledn</i> attribute includes the <i>cn=nsmanageddisabledrole, database suffix</i> role.
When synchronizing an object activation from Active Directory, Identity Synchronization for Windows activates the object by removing the <i>cn=nsmanageddisabledrole, database suffix</i> role from the object.	When synchronizing an object inactivation from Active Directory, Identity Synchronization for Windows inactivates the object by adding the <i>cn=nsmanageddisabledrole, database suffix</i> role to the object.

Modifying Directory Server's *NsAccountLock* Attribute Directly

Use this method when Directory Server activations and inactivations are based on Directory Server's operational attribute, *nsAccountLock*.

Note – When the Modify Directory Server's *nsAccountLock* attribute option is enabled, Identity Synchronization for Windows will not detect objects that are activated/inactivated using the Directory Server Console or command line utilities.

This attribute controls object states as follows:

- When `nsAccountLock=true`, the object is inactivated and the user cannot log in.
- When `nsAccountLock=false` (or has no value), the object is activated.

Modifying Directory Server's `nsAccountLock` Attribute Directly describes how Identity Synchronization for Windows detects and synchronizes object activations/inactivations when you enable the Modify Directory Server's `nsAccountLock` Attribute Directly option.

TABLE 4-2 Modifying Directory Server's `nsAccountLock` Attribute Directly

Activation	Inactivation
Identity Synchronization for Windows detects an inactivated object only when the <code>nsAccountLock</code> attribute is set to true .	Identity Synchronization for Windows detects an activated object only when the <code>nsAccountLock</code> attribute is absent or set to false .
When synchronizing an object inactivation from Active Directory, Identity Synchronization for Windows removes the <code>nsAccountLock</code> attribute.	When synchronizing an object activation from Active Directory, Identity Synchronization for Windows sets the <code>nsAccountLock</code> attribute to true .

Using a Custom Method for Directory Server

Use this method when Directory Server activations and inactivations are controlled exclusively by an external application such as Sun Java System Access Manager (formerly Sun JES Identity Server).

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

- How Identity Synchronization for Windows will detect that the external application has activated or inactivated an object in Directory Server.
- How Identity Synchronization for Windows will activate or inactivate the object when synchronizing from Active Directory to Directory Server.

Note – If you enable the Use custom method for Directory Server option, Identity Synchronization for Windows cannot lock objects out of the directory unless access to the directory is controlled by an external application, such as Access Manager.

To configure a Custom method for activations and inactivations, click the Configure button and the Configure Custom Method for Directory Server dialog box is displayed.

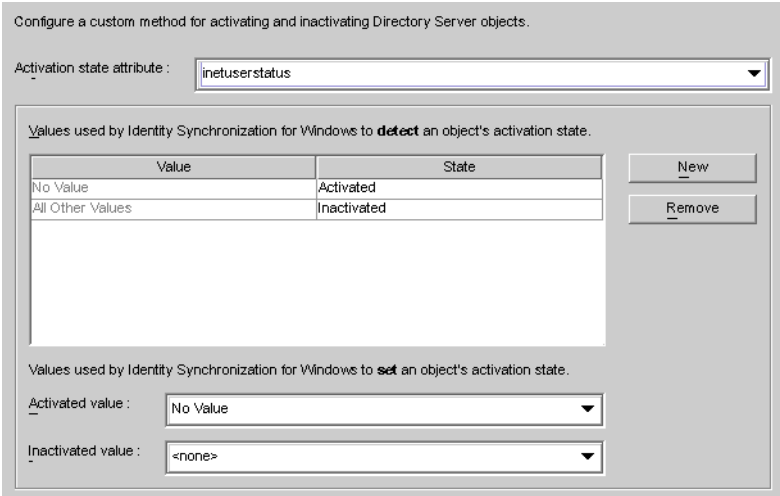


FIGURE 4–43 Configuring a Custom Method for Activations and Inactivations

This dialog contains the following features:

- **Activation state attribute drop-down list** : Use this list to specify an attribute that Identity Synchronization for Windows will use to synchronize activations and inactivations between Directory Server and Active Directory.
The list contains all attributes in the schema for the currently selected Directory Server structural and auxiliary objectclasses.
- **Value and State table**: Use this table to specify when values associated with the selected attribute are activated or inactivated.
 - **Value column**: Use this column (in conjunction with the New and Remove buttons) to specify attribute values that will be used to indicate active or inactive states.
The program automatically provides two values in this column:
 - **No Value**: Where the Activation state attribute has no value.
 - **All Other Values**: Where the Activation state attribute has a value, but that value is not specified in this Value and State table.
 - **State column**: Use this column to specify whether the Value entry (in the same row) corresponds to an object that is activated or inactivated.

Value	State	Result
No Value	Activated	If the attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as activated.

	Inactivated	If the attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as inactivated.
<i>user-defined values</i>	Activated	If the attribute has the <i>user-defined</i> attribute, Identity Synchronization for Windows detects the object as activated.
	Inactivated	If the attribute has the user-defined attribute, Identity Synchronization for Windows detects the object as inactivated.
All Other Values	Activated	If the attribute has a value, but that value is not specified in the table, Identity Synchronization for Windows detects the object as activated.
	Inactivated	If the attribute has a value, but that value is not specified in the table, Identity Synchronization for Windows detects the object as inactivated.

- **New button:** Click this button to add new entries to the Value column.
- **Remove button:** Select an entry in the Value column, and then click this button to remove that entry.
- **Activated value and Inactivated value drop-down lists:** Use these two lists to specify values that Identity Synchronization for Windows will use to *set* an object's state.

Synchronizing Activations and Inactivations

▼ To Configure Identity Synchronization for Windows to Detect and Synchronize Object States between Directory Server and Active Directory

- 1 Select an attribute from the Activation state attribute drop-down list.
- 2 Click the New button to add attribute values to the Value column of the table.
- 3 Click in the State column next to each of the Value entries and when the drop-down list is displayed, select Activated or Inactivated.

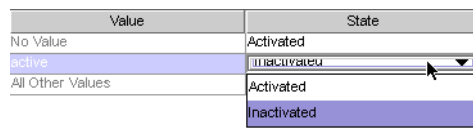


FIGURE 4-44 Selecting a State

For example, if you were using Access Manager:

- 4 Select the `inetuserstatus` attribute from the Activation state attribute drop-down list.
- 5 Click the New button and enter `active`, `inactive`, and `deleted` attribute values to the Value column of the table.
- 6 Click in the State column and select Activated or Inactivated for each value as follows:
 - **No Value:** Activated
 - **active:** Activated
 - **inactive:** Inactivated
 - **deleted:** Inactivated
 - **All Other Values:** Inactivated

Based on this example, “[Using a Custom Method for Directory Server](#)” on page 119 describes how Identity Synchronization for Windows will detect and synchronize activations/inactivations when you enable the Use Custom Method for Directory Server option (using the `inetuserstatus` example).

Value	State	Result
No Value	Activated	If the <code>inetuserstatus</code> attribute is missing or does not have a value, Identity Synchronization for Windows detects the object as activated.
active	Activated	If the attribute is active Identity Synchronization for Windows detects the object as activated.
inactive	Inactivated	If the attribute value is inactive Identity Synchronization for Windows detects the object as inactivated.
deleted	Inactivated	If the attribute value is deleted Identity Synchronization for Windows detects the object as inactivated.
All Other Values	Inactivated	If the attribute has a value, but that value is not specified in the table, Identity Synchronization for Windows detects the object as inactivated.

Setting Activations and Inactivations

As you populate the Value and State table with entries, Identity Synchronization for Windows automatically populates the **Activated value** and **Inactivated value** drop-down lists as follows:

- The Activated value list contains all values with an Activated status (for example **No Value** and **active**).
- The Inactivated value list contains all values with an Inactivated status (for example **inactive** and **deleted**).
- Neither list will contain the All Other Values value.

Select a value from the Activated value and/or the Inactivated value drop-down lists to specify how Identity Synchronization for Windows will activate and/or inactivate an object when synchronizing from Active Directory.

- **Activated value:** Controls the object's active state.
 - **No Value:** If the object contains the active value, Identity Synchronization for Windows will set the state to activated in Directory Server.
 - **active:** If the object contains the active value, Identity Synchronization for Windows will set the state to activated in Directory Server.
- **Inactivated value:** Controls the object's active state.
 - **inactive or deleted:** Identity Synchronization for Windows will set the object's state to inactive in Directory Server.
 - **none:** Not a valid setting. You must select a value.

Note – You must specify an Inactivated value or your configuration will be invalid.

Using a Custom Method for Directory Server illustrates a completed Configure Custom Method for Directory Server dialog box.

Configure a custom method for activating and inactivating Directory Server objects.

Activation state attribute : inetuserstatus

Values used by Identity Synchronization for Windows to **detect** an object's activation state.

Value	State
No Value	Activated
active	Activated
inactive	Inactivated
deleted	Inactivated
All Other Values	Inactivated

New

Remove

Values used by Identity Synchronization for Windows to **set** an object's activation state.

Activated value : No Value

Inactivated value : inactive

FIGURE 4-45 Example: Completed Dialog

Specifying Configuration Settings for Group Synchronization

If you enable Group Synchronization between Directory Server and Active Directory, you can synchronize the creation of groups, deletion of groups, and the membership changes within that group .

Note – Group Synchronization is not supported on Windows NT directory sources.

▼ To Synchronize Groups:

- 1 Under the Groups tab, select the Enable Group Synchronization check box.
- 2 Select one of the following Group Synchronization methods to specify how Identity Synchronization for Windows will detect and synchronize various groups:
 - *Domain Global Security*
 - *Domain Global Distribution*

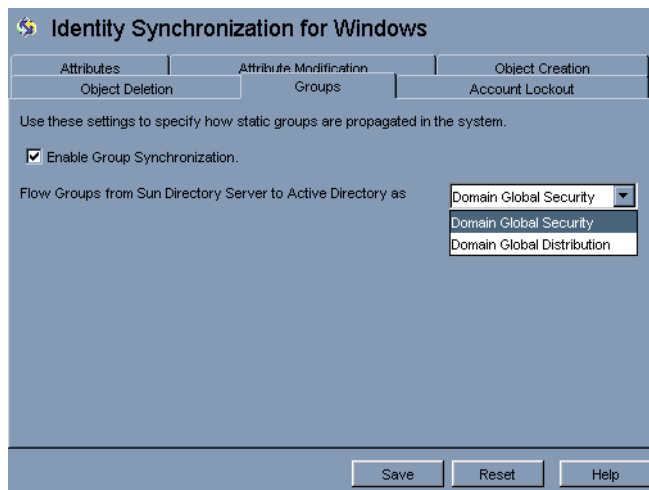


FIGURE 4–46 Enable Group Synchronization

Note – For more information about Domain Global Security, Domain Global Distribution, and Active Directory; see the Microsoft Active Directory documentation.

Configure Identity Synchronization for Windows to Detect and Synchronize Groups Related Changes between Directory Server and Active Directory

You do not need to map any attribute manually for the group synchronization. When you press Save, Identity Synchronization for Windows maps the attributes automatically.

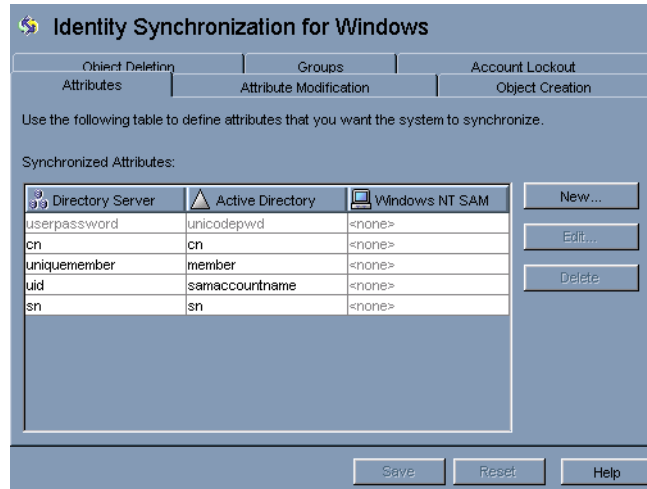


FIGURE 4-47 Attribute Mapping for Group Synchronization

Note –

1. Do not modify the mapping between the `userpassword` and `unicodepwd` attributes.
2. To disable the group synchronization, deselect the Disable Group Synchronization check box.
3. Alternatively, you can enable or disable group synchronization using command line `idsync groupsync`. For more information, see [Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities.”](#)

Configuring and Synchronizing Account Lockout and Unlockout

To enable the Account Lockout feature, you must do the following:

- Make the Password policies same on both Active Directory and Directory Server.

- Enable Account Lockout.
- Map certain attributes, which are different in Directory Server and in Active Directory

Identity Synchronization for Windows can synchronize the following events between Active Directory and Directory Server:

- Lockout events from Active Directory to Directory Server
- Lockout events from Directory Server to Active Directory
- Manual unlockout events from Active Directory to Directory Server
- Manual unlockout events from Directory Server to Active Directory

Note – Account lockout and unlockout synchronization is not supported on Windows NT directory servers.

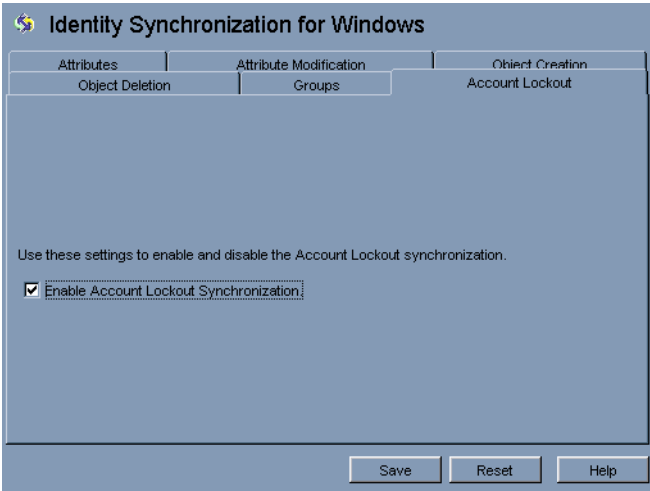
Prerequisites for Account Lockout

The attribute `lockoutDuration` should be set to the same value at both the places before enabling the account lockout feature. Make sure that the system time is also uniform across the distributed setup. Otherwise, the lockout events can expire if the `lockoutDuration` is less than the difference in the system dates.

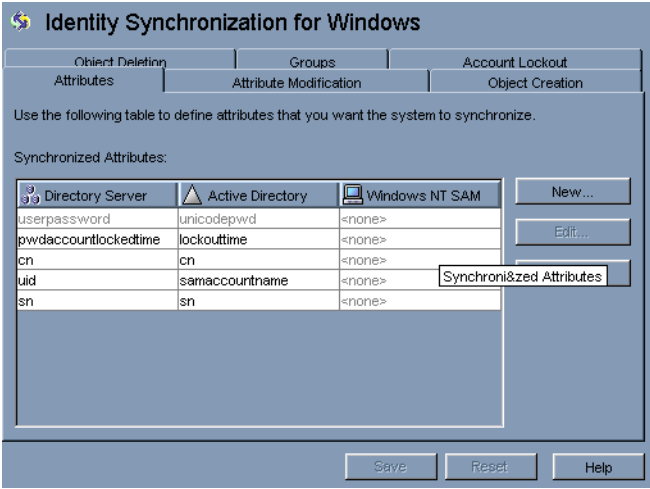
Note – Set the symmetric password policy at both ends. For example, if the password policy at Active Directory signifies a permanent lockout then the same password policy should be set at Directory Server.

Using the Account Lockout Feature

Enable Account Lockout Synchronization between Directory Server and Active Directory.



No explicit mapping of the `pwdaccountlockedtime` (Directory Server) and `lockoutTime` (AD) attributes is required to enable account lockout. Select **Enable Account Lockout Synchronization** from the **Account Lockout** tab in **Identity Synchronization for Windows** configuration panel.



Note – You can enable or disable the account lockout synchronization using command line tool **idsync accountlockout**. For more information, see [Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities.”](#)

Specifying How Deletions Flow

Use Object Deletions tab to specify how deleted user entries should flow between Directory Server and Active Directory systems.

Note – You cannot specify Object Deletions flow for Windows NT.

▼ To Specify how Deleted Entries Flow Between Directory Server and Active Directory Systems

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

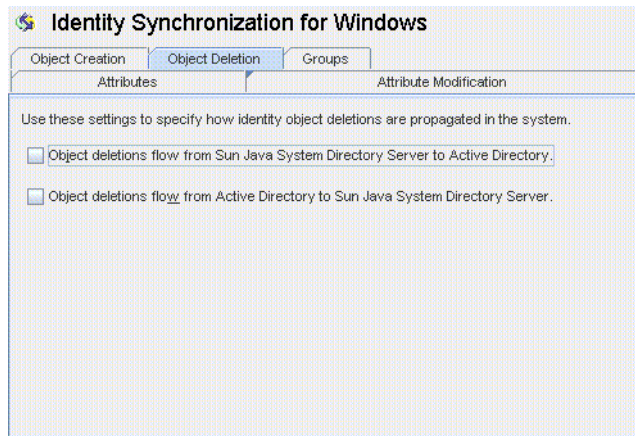


FIGURE 4–48 Propagating User Entry Deletions

- 2 Enable or disable the flow of deletions as follows:
 - Enable **Object deletions flow from Sun Java System Directory Server to Active Directory** to propagate deletions from the Sun Directory Server environment to your Active Directory servers.
 - Enable **Object deletions flow from Active Directory to Sun Java System Directory Server** to propagate deletions from the Active Directory environment to your Sun Directory Servers.
 - Enable both options for bidirectional flow.
 - Disable both options to prevent user deletions from propagating from one system to the other (*Default setting*).

Creating Synchronization User Lists

A Synchronization User List (SUL) specifies which users in Active Directory and Sun Directory Server will be synchronized. Every entry in the SUL passes through the Connector and is evaluated against the constraints you configured for that SUL.

Each SUL contains two elements, one to identify which Directory Server users to synchronize and one to identify which Windows users to synchronize.

Note – To synchronize users in a Directory Server with multiple Active Directory domains, you must define one SUL for each Active Directory domain.

For more information about defining and configuring SULs (including components of a definition, how to define multiple SULs, how multiple SULs are processed, and how to configure multiple Windows domain support) refer to [Appendix D, “Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows”](#)

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.

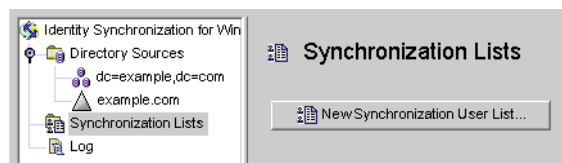


FIGURE 4-49 Creating a New Synchronization User List

The Define a Synchronization User List wizard is displayed.

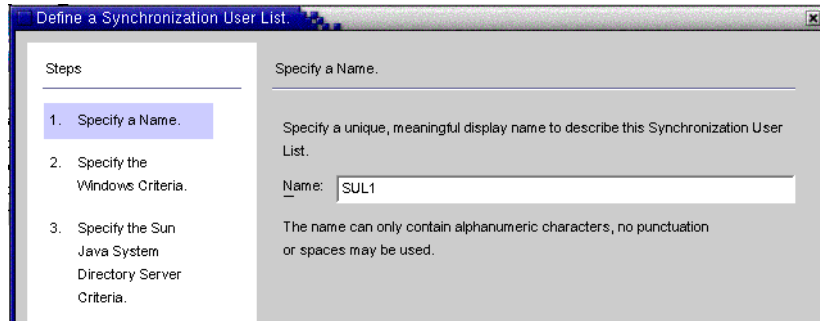


FIGURE 4-50 Specifying a Name for Your SUL

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

- If the default name is acceptable, click Next.
- If you want to use a different name, type a different name into the Name field and then click Next.
- Do not use spaces or any kind of punctuation in the SUL name.
- You must specify a name that is unique within the system.

The Windows Criteria panel is displayed.

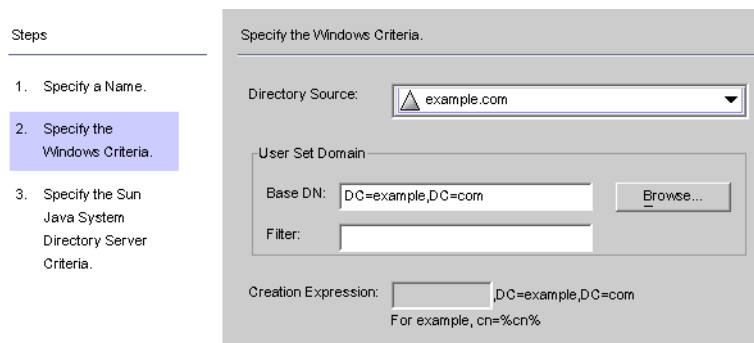


FIGURE 4-51 Specifying the Windows Criteria

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

Note – You cannot edit the Active Directory or Directory Server directory sources included in this SUL after you click the Finish button to create the SUL. When the Group Synchronization feature is enabled, the creation expression would be `uid=%uid%` or `cn=%cn%` in the Sun Java System Directory Server Criteria panel.

3 A User Set Domain is the set of all the users to be synchronized. Enter the User Set Domain's Base DN, using one of the following methods:

- Type the name into the text field (for example, **DC=example,DC=com**).
- Click the Browse button, to open the Set Base DN dialog box so you can look for, and select a Base DN.

All users under the specified Base DN will be included in this SUL, unless you explicitly exclude them using a filter.

Note – Base DNs and creation expressions are not allowed for Windows NT machines.

You cannot edit the Active Directory or Directory Server directory sources included in this SUL after you click the Finish button to create the SUL. When the Group Synchronization feature is enabled, then the creation expression should be `uid=%uid%` in the Sun Java System Directory Server Criteria panel.

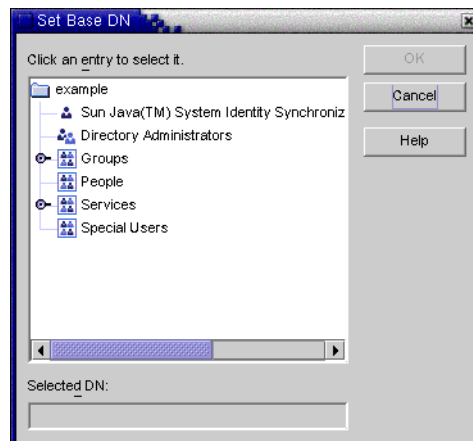


FIGURE 4-52 Selecting a Base DN

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. For more information, see [“Specifying How Object Creations Flow” on page 111](#).

- 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.

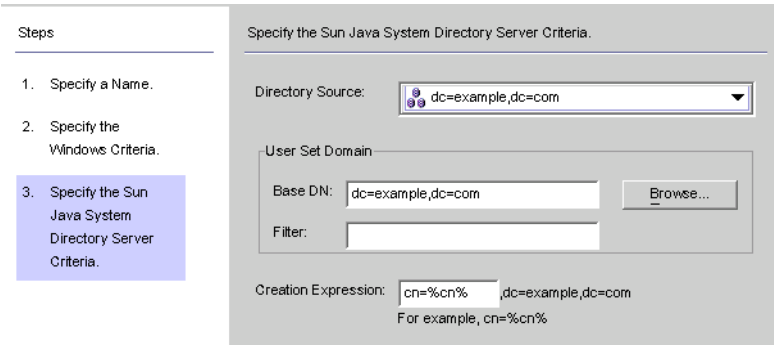


FIGURE 4–53 Specifying Directory Server Criteria

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.

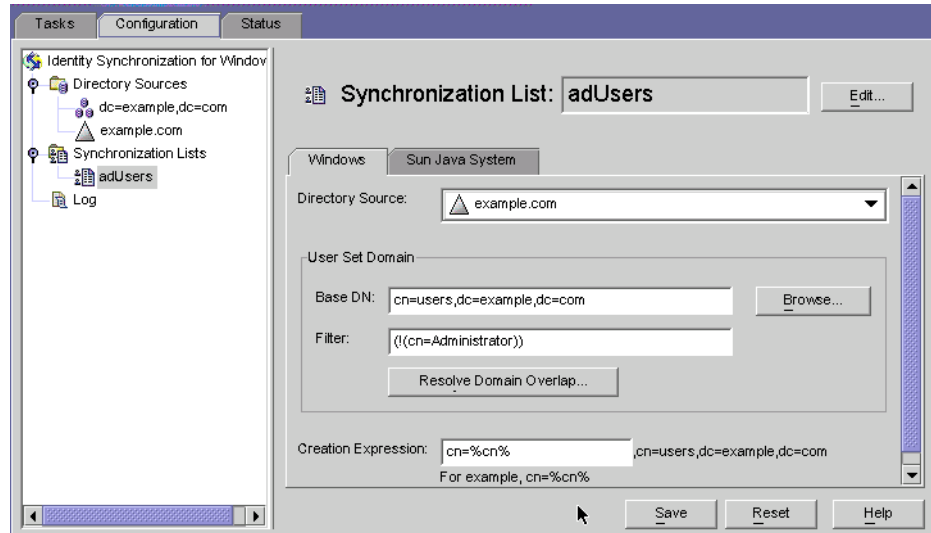


FIGURE 4-54 Synchronization List Panel

- 10 In cases where a user matches multiple lists, click the **Resolve Domain Overlap** button to define a preference for the synchronization user list.
- 11 Create a **Synchronization User List** that includes every directory source in your network except for the **Directory Server**.

Saving a Configuration

▼ To Save your Current Configuration from the Console Panels

- 1 Click **Save** to store your settings at this point.
- 2 The **Configuration Validity Status** window is displayed as the program evaluates your configuration settings.

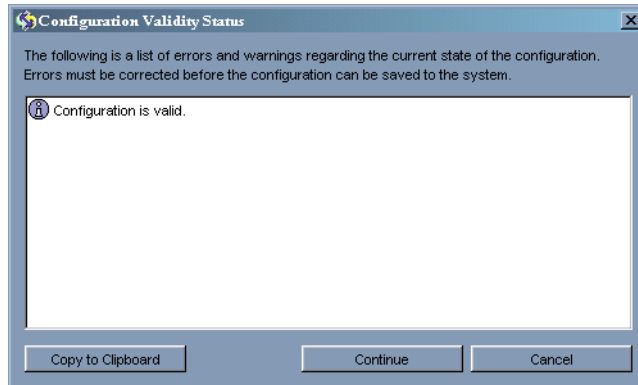


FIGURE 4-55 Configuration Validity Status Window

This panel confirms that your configuration is valid or identifies configuration problems that must be fixed.

Saving your configuration may take a few minutes because the program rewrites the information out to the configuration directory and notifies the system manager.

The system manager (a Core component) is responsible for distributing your configuration settings out to the components that need the information.

Note – Configuration validation errors are red and warnings are yellow.

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

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

A Connector Installation Instructions dialog box is displayed, giving instructions about how to proceed with installing the Identity Synchronization for Windows Connectors and subcomponents.

This list has now been updated with a To Do list that is customized for your deployment. (Up to this point, the steps were generic.) Note that you can also access and update the To Do list from the Status tab on the Identity Synchronization for Windows Console.

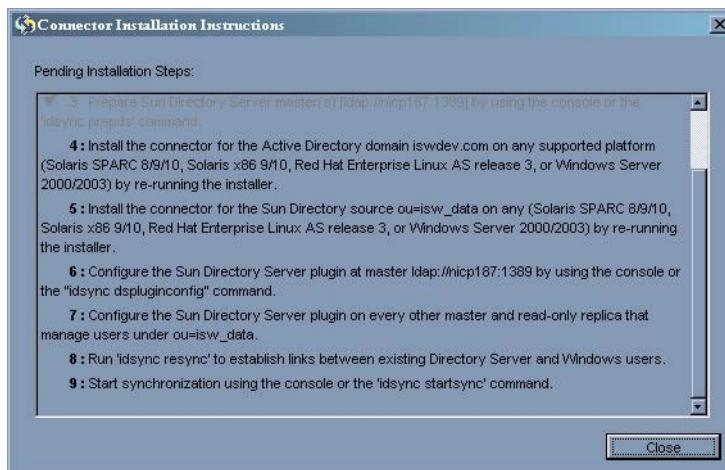


FIGURE 4-56 Instructions for Installing the Connectors

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 1, “Understanding the Product,”](#) for instructions.

Installing Connectors

This chapter provides instructions for installing the Identity Synchronization for Windows Connectors. The information is organized as follows:

- [“Before You Begin” on page 137](#)
- [“Running the Installation Program” on page 138](#)
- [“Installing Connectors” on page 140](#)

Identity Synchronization for Windows uses Connectors to synchronize user passwords between directory sources, and uses subcomponents to enhance the Connector’s change-detection and bidirectional synchronization support.

Before You Begin

Before starting the Connector configuring process, you should be aware of the following:

- Close the Console before starting the installation process. If the Console is open when you are installing a Connector, the program perceives a conflict about which component is adding configuration data to the server and generates an error message.
- Active Directory Connectors do not have subcomponents.
- Windows NT Connectors and subcomponents are installed simultaneously.
- You can install Directory Server or Active Directory Connectors on the same machine where you installed Core or you can install Connectors on another machine. (The Windows NT Connector must be installed on the Primary Domain Controller (PDC) of the domain being synchronized.)
- If you are installing the Connector on the same machine as Core, the program automatically installs the Connector in the same directory as Core.
- If you are installing the Connector on a different machine, the program will prompt you to specify the configuration directory information supplied during the Core installation.

You must run the installation program each time you install a Connector.

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

Running the Installation Program

Repeat the following steps each time you install a Connector.

▼ To Restart and Run the Installation Program

- 1 **Run the installation program again 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 Linux:** Change to the `installer` directory and then type `./installer.sh` to execute the installation program.

Note – To run the installation program in text-based mode, type `./installer.sh -nodisplay`.

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

- **On Windows:** Change to the `installer` directory and then type `setup.exe` to execute the installation program.
- 2 **When the Welcome screen is displayed, read the information provided and then click Next to proceed to the Software License Agreement panel.**
 - 3 **Read the license agreement, then select**
 - **Yes (Accept License)** to accept the license terms and go to the next panel.
 - **No** to stop the setup process and exit the installation program.

4 The Sun Java System Directory Server panel is displayed. Specify the configuration directory location as follows:

- **Configuration Directory Host:** Enter the fully qualified domain name (FQDN) of a Sun Java System Directory Server instance (affiliated with an Administration Server) where Identity Synchronization for Windows configuration information is stored. You must specify the same instance that you specified during the Core installation.
- **Configuration Directory Port** (Defaults to port 389): Specify a port for the configuration directory. You can leave the port set to the default or change to a different, available port.
To enable SSL (Secure Socket Layer) between Core and the configuration directory, enable the Secure Port option and specify an SSL port (*default SSL port is 636*). Enabling this option prevents sensitive information from being passed in the clear over the network.
- **Configuration Root Suffix:** Select the root suffix that you specified during the Core installation from the menu. The Identity Synchronization for Windows configuration will be stored in this root suffix.

Note – If the program could not detect a root suffix, and you enter the server information manually, you must click Refresh to repopulate the list of root suffixes.

5 Click Next to open the Configuration Directory Credentials panel.

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

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

Note – These credentials will be sent without encryption unless you enabled SSL in.

7 Click Next to open the Configuration Password panel where you must enter the configuration password you specified when you installed Core.

Also, if Core has not been installed on this machine, you will be prompted to provide the location of the Java Home directory (see [“Installing Core” on page 70](#)).

8 When you are finished, click Next.

Note – At this point, the installation process becomes specific to the type of Connector you are installing.

Installing Connectors

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

- “Installing the Directory Server Connector” on page 140
- “Installing an Active Directory Connector” on page 146
- “Installing the Windows NT Connector” on page 148

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 138

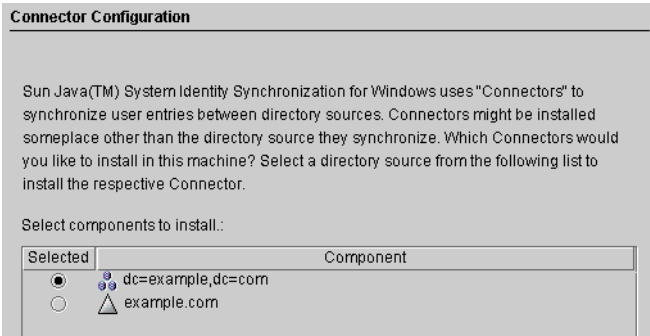


FIGURE 5-1 Selecting the Directory Server Connector

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), 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

TABLE 5-1 Directory Source Examples (Continued)

Directory Source	Example Entry
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.

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.

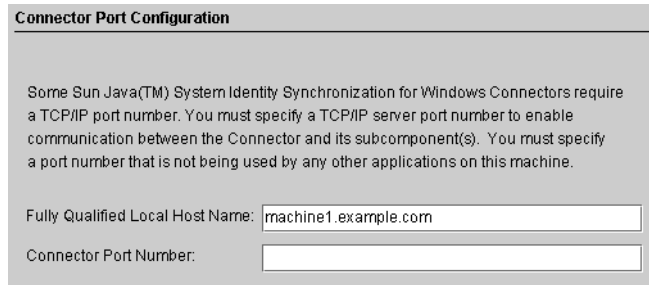
Enter the following information:

- **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.

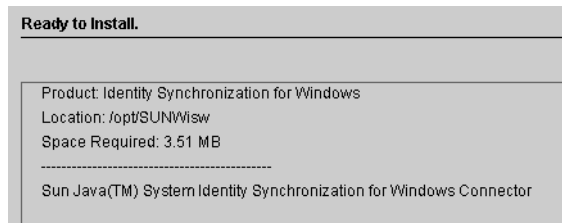
The program will verify that the Directory Server was prepared and ready to synchronize data. When you prepared Directory Server ([“Preparing Sun Directory Source” on page 91](#)), the program creates an account that the Connector will use to connect to Directory Server (for example, `uid=PSWConnector,suffix`).

2 Click Next to proceed to the Connector Port Configuration pane.



The screenshot shows a window titled "Connector Port Configuration". Inside, there is a paragraph of text: "Some Sun Java(TM) System Identity Synchronization for Windows Connectors require a TCP/IP port number. You must specify a TCP/IP server port number to enable communication between the Connector and its subcomponent(s). You must specify a port number that is not being used by any other applications on this machine." Below this text are two input fields. The first is labeled "Fully Qualified Local Host Name:" and contains the text "machine1.example.com". The second is labeled "Connector Port Number:" and is empty.

- 3 Enter the Fully Qualified Local Host Name with the domain and an available port number where the Connector will listen. (Specifying a port already in use will result in an error message.)**
- 4 Click Next and the Ready to Install pane is displayed to provide information about the Connector's installation location and how much disk space is required for the installation. When you are ready, click the Install Now button.**



The screenshot shows a window titled "Ready to Install." Below the title bar, there is a list of installation details: "Product: Identity Synchronization for Windows", "Location: /opt/SUNWisiw", and "Space Required: 3.51 MB". A horizontal dashed line separates this information from the text "Sun Java(TM) System Identity Synchronization for Windows Connector" at the bottom.

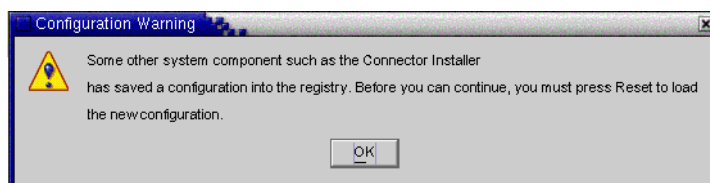
Note – If you installed Core on the local machine, the Ready to Install pane will indicate that zero space is required to install the Connector. This situation occurs because the Core installation has already installed the Connector binaries. Because there are no additional binaries to install, no additional space is required.

If you are installing the Connector on a machine other than where you installed Core, then the Ready to Install pane will indicate how much space is required to complete the Connector installation on the local machine.

The Connector installation is accomplished in two steps:

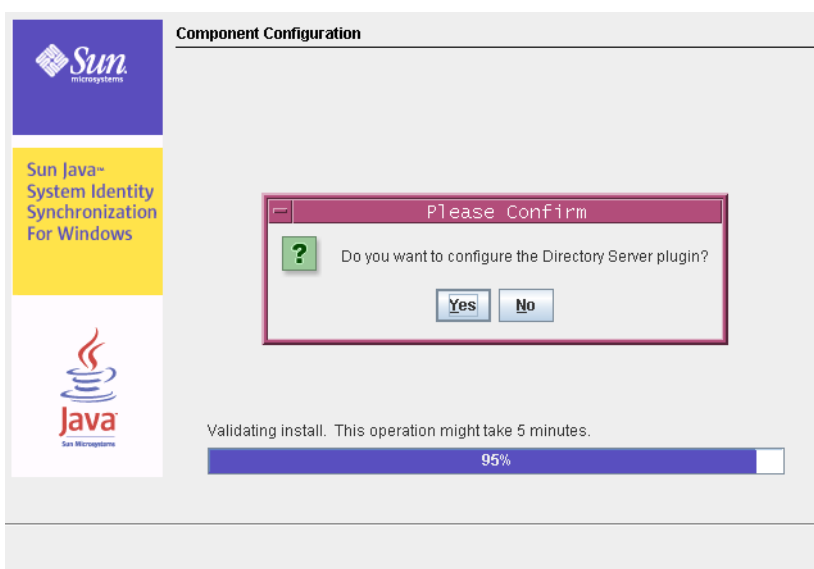
- An Installing pane is displayed, with a progress bar, while the program installs the binaries.
- Next, the Component Configuration pane displays a progress bar. This step takes several minutes to complete.

Note – If you did not close the Console before starting the installation, the following warning displays (“[Installing the Directory Server Connector](#)” on page 140). Click Reset in the Console to reload the Connector’s configuration settings.



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

Note – Directory Server plugin gets configured for preferred and secondary hosts (if any).



Note –

- a. Clicking Yes configures the Directory Server plugin in all the hosts (preferred and secondary).
- b. Clicking No enables you to configure the plugin later using command line **idsync dspluginconfig**. For more information, see [Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities.”](#)

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

- **On Solaris:** Installation logs are written to /var/sadm/install/logs/
- **On Linux:** Installation logs are written to /var/sadm/install/logs/
- **On Windows:** Installation logs are written to the %TEMP% directory, which is usually a subdirectory of the Local Settings folder located 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 to display the “To Do list” panel, which shows the list of successfully completed and pending steps.

This is a list of remaining installation and configuration steps:

- ✓ 1 : Install the Identity Synchronization core components.
- ✓ 2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importcnf'.
- ✓ 3 : Prepare Sun Directory Server master(s) [ldap://isw080:1389] by using the console or the 'idsync prepds' command.
- ✓ 4 : Install the connector for the Sun Directory source ou=isw_data on any (Solaris SPARC 8/9/10, Solaris x86 9/10, Red Hat Enterprise Linux AS release 3, or Windows Server 2000/2003) by re-running the installer.
- ✓ 5 : Configure the Sun Directory Server plugin at master ldap://isw080:1389 by using the console or the "idsync dspluginconfig" command.
- 6 : Install the connector for the Active Directory domain metaqa.com on any supported platform (Solaris SPARC 8/9/10, Solaris x86 9/10, Red Hat Enterprise Linux AS release 3, or Windows Server 2000/2003) by re-running the installer.
- 7 : Configure the Sun Directory Server plugin on every other master and read-only replica that manage users under ou=isw_data.
- 8 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.
- 9 : Start synchronization using the console or the 'idsync startsync' command.

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

After installing the Directory Server Connector, you can install other Connectors that you configured when you configured the resources ([Chapter 4, “Configuring Core Resources”](#)):

- Install additional Directory Server Connectors: Restart the installation program (using the instructions in [“Running the Installation Program” on page 138](#)) and then repeat [Step 1](#) through [Step 7](#).
- Install an Active Directory Connector: Go to [“Installing an Active Directory Connector” on page 146](#).
- Install a Windows NT Connector: Go to [“Installing the Windows NT Connector” on page 148](#).

Configuring Identity Synchronization for Windows Plug-in when Chained Suffix exists

This configuration is needed only when the chained suffix exists in the Directory Server instance where Identity Synchronization for Windows Plug-in is installed. If Identity Synchronization for Windows Plug-in is not configured to search on chained suffix, MODIFY and BIND operations performed on the Directory Server where the Identity Synchronization for Windows Plug-in is installed, will fail.

In the Directory Server instance where the chained suffix is created, perform the following operations:

Execute the following LDIF script using `ldapmodify` utility:

```
dn: cn=config,cn=chaining database,cn=plugins,cn=config
changetype: modify
add: nspossiblechainingcomponents
nspossiblechainingcomponents: cn=pswsync,cn=plugins,cn=config
```

You can perform the similar operation by using the following procedure:

1. Select the Configuration tab.
2. Click the Data node that displays in the left pane.
3. Select the Chaining tab in the right pane.
4. Add Identity Synchronization for Windows Plug-in (cn=pswsync, cn=plugins, cn=config) to the components that are allowed to chain.
5. Save the changes and exit.

Installing an Active Directory Connector

After you install the Directory Server Connector and if you have other configured Connectors to install, the installation program will give you the option of installing the Connectors before you see the Connector Configuration pane.

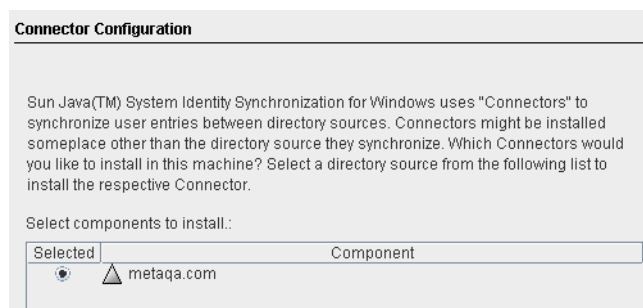


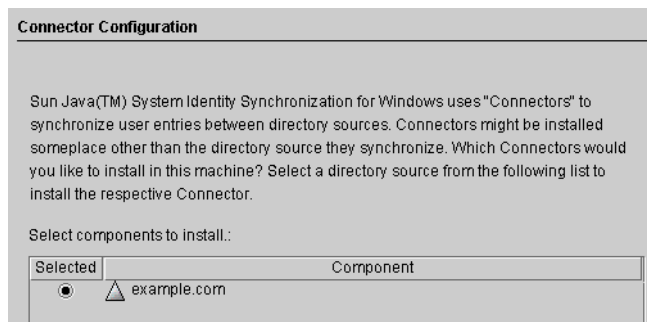
FIGURE 5-2 Selecting the Connector

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

▼ To Install an Active Directory Connector

1 Enable the Connector button and click Next.

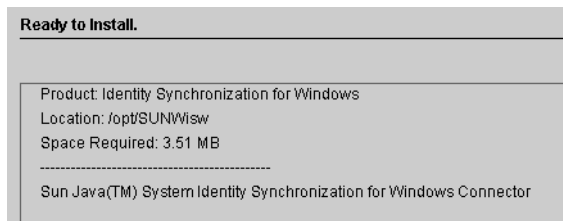
The Connector Configuration panel displays.



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

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

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



Note – If you installed Core on the local machine, the Ready to Install pane will indicate that zero space is required to install the Connector. This situation occurs because the Core installation has already installed the Connector binaries. Because there are no additional binaries to install, no additional space is required.

If you are installing the Connector on a machine other than where you installed Core, then the Ready to Install pane will indicate how much space is required to complete the Connector installation on the local machine.

3 When you are ready, click the Install Now button.

An Installing pane is displayed, with a progress bar, while the program installs the binaries, and then an Installation Summary pane is displayed to confirm the installation is finished.

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

- **On Solaris:** Installation logs are written to /var/sadm/install/logs/
- **On Linux:** Installation logs are written to /var/sadm/install/logs/
- **On Windows:** Installation logs are written to the %TEMP% directory, which is a subdirectory of the Local Settings folder located 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 to display the "To Do list" panel, which shows the list of successfully completed and pending steps.

This is a list of remaining installation and configuration steps:

- ✓ 1 : Install the Identity Synchronization core components.
- ✓ 2 : Create an initial configuration using the product's console or by migrating from a previous installation using 'idsync importcnf'.
- ✓ 3 : Prepare Sun Directory Server master(s) [ldap://isw080:1389] by using the console or the 'idsync prepds' command.
- ✓ 4 : Install the connector for the Active Directory domain metaqa.com on any supported platform (Solaris SPARC 8/9/10, Solaris x86 9/10, Red Hat Enterprise Linux AS release 3, or Windows Server 2000/2003) by re-running the installer.
- ✓ 5 : Install the connector for the Sun Directory source ou=isw_data on any (Solaris SPARC 8/9/10, Solaris x86 9/10, Red Hat Enterprise Linux AS release 3, or Windows Server 2000/2003) by re-running the installer.
- ✓ 6 : Configure the Sun Directory Server plugin at master ldap://isw080:1389 by using the console or the "idsync dspluginconfig" command.
- 7 : Configure the Sun Directory Server plugin on every other master and read-only replica that manage users under ou=isw_data.
- 8 : Run 'idsync resync' to establish links between existing Directory Server and Windows users.
- 9 : Start synchronization using the console or the 'idsync startsync' command.

6 When you are done with the panel, click Finished to exit the installation program.

After installing the Active Directory Connector, you can install other Connectors that you configured when you configured resources ([Chapter 4, “Configuring Core Resources”](#)):

- Install additional Active Directory Connectors: Restart the installation program (see [“Running the Installation Program” on page 138](#)) and then repeat through.
- Install a Windows NT Connector: Go to [“Installing the Windows NT Connector” on page 148](#).
- Install additional Directory Server Connectors: Restart the installation program (using the instructions in [“Running the Installation Program” on page 138](#)) and then repeat [Step 1](#) through [Step 6](#).

Installing the Windows NT Connector

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

▼ To Install a Windows NT Connector and the NT subcomponents

- 1 Enable the Windows NT Connector button and click Next.
- 2 When the Connector Port Configuration pane is displayed, enter the Fully Qualified Local Host Name with the domain and an available port number where the Connector will listen. (Specifying a port already in use will result in an error message.)

3 When you are done, click Next.

The Ready to Install pane is displayed to provide information about the Connector's installation location and how much disk space is required.

4 When you are ready, click the Install Now button.

The Connector installation is accomplished in two steps:

- An Installing pane is displayed, with a progress bar, while the program installs the binaries.
- Next, the Component Configuration pane displays a progress bar. This step takes several minutes to complete.

Note – If you did not close the Console before starting the installation, a warning displays (see [“Installing the Directory Server Connector” on page 140](#)). Click Reset in the Console to reload the Connector's configuration settings.

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

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

Installation logs are written to the %TEMP% directory, which is C:\TEMP on most Windows NT systems.

6 Click Close to exit the installation program.

After installing the Windows NT Connector, you can install other Connectors that you configured when you configured resources ([Chapter 4, “Configuring Core Resources”](#)):

- To install additional Windows NT Connectors, restart the installation program. For more information, see [“Running the Installation Program” on page 138](#) and then repeat [Step 1](#) through [Step 6](#).
- To install Directory Server Connector, refer to [“Installing the Directory Server Connector” on page 140](#).
- To install Active Directory Connector, refer to [“Installing an Active Directory Connector” on page 146](#).

Synchronizing Existing Users and User Groups

The Identity Synchronization for Windows command line utility provides the `idsync resync` subcommand to bootstrap deployments with existing users or groups. This command uses administrator-specified matching rules to link existing entries, to populate an empty directory with the contents of a remote directory, or to bulk-synchronize attribute values (including passwords) between two existing user and group populations.

This chapter explains how to use the `idsync resync` subcommand and synchronize existing users and groups for new Identity Synchronization for Windows installations. In addition, this chapter provides instructions for starting and stopping synchronization and services. The information is organized as follows:

- “Using `idsync resync`” on page 152
- “Checking Results in the Central Log” on page 157
- “Starting and Stopping Synchronization” on page 158
- “Starting and Stopping Services” on page 159

Note – You must finish installing Core and the Connectors before trying to synchronize existing users.

For more information about the `idsync resync` subcommand, see [Appendix A, “Using the Identity Synchronization for Windows Command Line Utilities”](#)

Synchronizing Existing Users and User Groups summarizes the post-installation steps to follow based on existing user and group populations:

Post-Installation Steps Based on Existing User and Group Populations

TABLE 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 <code>idsync resync -o Sun -c</code> to create existing Directory Server users in Windows.	None
Yes	No	Run <code>idsync resync -c</code> to create existing Windows users in Directory Server.	Run <code>idsync resync -u</code> to populate the connector's local cache of user entries.
Yes	Yes	Run <code>idsync resync -f <filename> -k</code> to link the users only, and then run <code>idsync resync -o Sun</code> to resynchronize existing users from Directory Server.	Run <code>idsync resync -u</code> to populate the connector's local cache of user entries.

Note – If Group Synchronization is enabled then the groups are synchronized in the same way as the users are synchronized.

Using idsync resync

This section explains the synchronizing processes, describes the proper syntax for using the `idsync resync` subcommand, and explains how to verify that the processes completed successfully. The information is organized as follows:

- [“Resynchronizing Users or Groups” on page 152](#)
- [“Linking Users” on page 153](#)
- [“idsync resync Options” on page 154](#)
- [“Checking Results in the Central Log” on page 157](#)

Resynchronizing Users or Groups

You need to resynchronize the user entries when two directory sources become out of sync. Use the `idsync resync` command to create users, user groups, and synchronize user and user group

attributes in two directory sources. Specifically, you can use the `idsync resync` command to populate an empty Directory Server with the existing Active Directory or Windows NT SAM domain users.

The `idsync resync` command can be used in any of the following ways:

- If there are users that exist on Directory Server and Windows, you must run the `idsync resync` command to synchronize those users.
- If you do not want to synchronize existing users to Directory Server, then run `idsync resync` with the `-u` argument, which updates the object cache only and does not synchronize the Windows' entries to Directory Server.
- If you have existing Windows users and do not run `idsync resync`, then changes to these users may or may not be propagated; and depending on flow settings, these users might even be automatically created in Directory Server. You must run `idsync resync` again, even if you have already run the command.

Note – You cannot use the `idsync resync` command to synchronize passwords (except to invalidate Directory Server passwords to force on-demand password synchronization in an Active Directory environment).

When the Group Synchronization feature is enabled, both the users as well as the groups associated with the users are synchronized between the data sources configured. No additional options are required while using the `resync` command for Group Synchronization.

Linking Users

After populating Active Directory and Directory Server with users and installing the Active Directory and Directory Server Connectors (before starting synchronization), you must use the `idsync resync` command to ensure that all existing users are *linked* in the two directory sources.

What is *linking*? Identity Synchronization for Windows correlates the same user on Directory Server and on Windows by storing the following unique, immutable identifiers:

- The `dspswuserlink` attribute of each Directory Server user entry
- The `objectguid` attribute for each Active Directory user
- A combination of the domain name and the RID for each Windows NT SAM user

Storing this immutable identifier allows Identity Synchronization for Windows to synchronize other key identifiers, such as `uid` and `cn`. The `dspswuserlink` attribute is populated when:

- Identity Synchronization for Windows creates a new user in Directory Server (after a new user is synchronized from Windows or by 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.

Note – If Group Synchronization is enabled then the groups are linked in the same way as the users are linked.

idsync resync Options

The `idsync resync` command accepts the following options.

TABLE 6-2 idsync resync Usage

Argument	Meaning
-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 <code>idsync resync -o Sun -a "uid=*"</code> 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 <ul style="list-style-type: none"> ■ 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 <ul style="list-style-type: none"> ■ Randomly generates a cryptographically secure password for users created in Active Directory or Windows NT. ■ Automatically creates a special password value ({PSWSYNC} *INVALID PASSWORD*) for users created in Directory Server (unless you specify the -i option) Note: Identity Synchronization for Windows will attempt to create users even if you have not configured creations in that direction. For example, if you have not configured Identity Synchronization for Windows to synchronize from Windows to Sun (or vice versa), but you specify the -c argument, Identity Synchronization for Windows will try to create users that are not found.
-i (ALL_USERS NEW_USERS)	Resets passwords for user entries synchronized in a Sun directory source, forcing password synchronization within the current domain for those users the next time the user password is required. <ul style="list-style-type: none"> ■ ALL_USERS: Forces on-demand password synchronization for all synchronized users ■ NEW_USERS: Forces on-demand password synchronization for newly created users only

TABLE 6-2 idsync resync Usage (Continued)

Argument	Meaning
-u	Updates the object cache. This argument updates the local cache of user entries for a Windows directory source only, which prevents 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_USERS	No	No	Yes
No -i value	No	No	No

The following table provides examples to illustrate the results of combining different arguments (The -h, -p, -D, -w, -, and -s arguments are defaulted and have been omitted for brevity).

TABLE 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.

TABLE 6-4 idsync resync Usage Samples (Continued)

Arguments	Result
idsync resync	Displays a resync usage statement.
idsync resync -c -l SUL_sales -l SUL_finance	Creates all existing Active Directory users on Directory Server for the SUL_sales and SUL_finance SULs only (but does not force on-demand password synchronization).
idsync resync -n	Runs in safe mode so you can preview the effects of the resync operation with no actual changes.
idsync resync -o Sun -a "(sn=Smith)"	Synchronizes all Directory Server users with the last name (sn) Smith, on Windows.
idsync resync -u	Updates the object cache for Windows Connectors only to prevent existing users from being created in Directory Server. No users are actually synchronized.
idsync resync -f link.cfg	Links unlinked users based on linking criteria specified in the link.cfg file. Identity Synchronization for Windows does not create or modify users, but the Directory Server passwords of newly linked users will be set to the Active Directory users' passwords.

Note – When you use `idsync resync` to link users, be aware that you should use indexes for the operation. Non-indexes 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 are no indexes in `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.

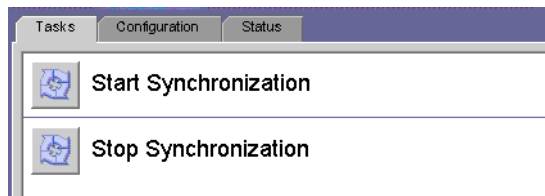


FIGURE 6-1 Starting and Stopping Synchronization

- 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 219](#) and [“Using stopsync” on page 220](#)

Resynchronized Users/Groups

To resynchronize groups, the Group Synchronization feature must be enabled either through the console or through the command line interface.

To know about how to enable the Group Synchronization feature, see [“Specifying Configuration Settings for Group Synchronization” on page 124](#)

Starting and Stopping Services

Identity Synchronization for Windows and Message Queue are installed as *daemons* on Solaris and Linux, and as *services* on Windows. These processes start automatically when the system boots, but you can also start and stop them manually, as follows:

- **On Solaris:** From the command line,
 - Enter `/etc/init.d/isw start` to start all Identity Synchronization for Windows processes.
 - Enter `/etc/init.d/isw stop` to stop all Identity Synchronization for Windows processes.
 - Enter `/etc/init.d/imq start` to start the Message Queue broker.
 - Enter `/etc/init.d/imq stop` to stop the Message Queue broker.
- **On Linux:** From the command line,
 - Enter `/etc/init.d/isw start` to start all Identity Synchronization for Windows processes.
 - Enter `/etc/init.d/isw stop` to stop all Identity Synchronization for Windows processes.
 - Enter `/etc/init.d/imq start` to start the Message Queue broker.
 - Enter `/etc/init.d/imq stop` to stop the Message Queue broker.
- **On Windows:**
 - From the Windows Start menu:
 1. Select Start → Settings → Control Panel → Administrative Services.
 2. When the Administrative Services dialog box is displayed, double-click the Services icon to open the Services dialog box.
 3. Select Identity Synchronization for Windows and then select Action → Start (or Stop) from the menu bar. Repeat for iMQ Broker.
 - From the command line, enter the `net` command to control the services.

Note – Pause 30 seconds after stopping the Identity Synchronization for Windows daemon/service before starting it again. Connectors can take several seconds to cleanly shut themselves down.

Removing the Software

This section contains procedures for removing Identity Synchronization for Windows 6.0 in the following sections:

- [“Planning for Uninstallation” on page 161](#)
- [“Uninstalling the Software” on page 162](#)
- [“Uninstalling the Console Manually” on page 166](#)

Planning for Uninstallation

Before removing the software keep in mind the following points:

- You must uninstall subcomponents and the Directory Server Plug-in before you uninstall their associated connectors, and uninstall all the connectors before Core. (The Active Directory Connector does not have any subcomponents to uninstall.)
Failure to uninstall one of these components in the proper order will prevent you from selecting and uninstalling the other components. For example, if you do not uninstall the connectors first, you cannot select Core for uninstallation.
- You must uninstall the Directory Server Plug-in before you uninstall Core.
Uninstalling Core first will remove the Plug-in bits without unregistering them from the Directory Server, which prevents the Directory Server from starting unless you manually remove `cn=pswsync`, `cn=plugins`, `cn=config`.
- In replicated environments with replicas (in addition to primary and secondary servers) you must uninstall the Directory Server Plug-in and then restart the servers.
- The order in which you uninstall connectors does not matter.
- After uninstalling a Sun Java System Directory Server or Windows Connector, you must perform some additional steps to reinstall the Connector on a different machine or to use a different server port.

In this case, you must uninstall and reinstall all of the corresponding subcomponents, and restart the Identity Synchronization for Windows daemon/service where Core is installed (see [“Starting and Stopping Services” on page 159](#)).

- You must run the `uninstall.cmd` script (located in the `isw-hostname` directory) on Windows 2000 and NT platforms. (You must run this batch file as Administrator.)
- You must run the `runUninstall.sh` script (located in the installation directory, `/opt/SUN/isw`, by default) on the Solaris or Linux operating systems. (You must run this script as root.)

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

Uninstalling the Software

Your system may contain any or all of the following Identity Synchronization for Windows components:

- Active Directory Connectors
- Directory Server Connectors and Plug-ins
- Core

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

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

This section provides instructions for the following:

- [“Uninstalling Connectors” on page 162](#)
- [“To Uninstall Core” on page 164](#)

Uninstalling Connectors

▼ To Uninstall the Connectors

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

These programs are located in the installation directory (which is the `/opt/SUNWisw` directory by default).

- 2 **At the Welcome screen click Next.**

3 Enter the Configuration Directory Host name and Port number.

- Select the root suffix of the configuration directory. (If necessary, click Refresh to see the list of suffixes.)
- For secure communication between the uninstall program and the configuration directory server, enable the Secure Port box and specify the Directory Server's SSL port number.

4 Enter your administrator's name and password for the configuration directory.**5 Select the connector(s) to be uninstalled.**

Note – The selected connectors *must* be present on the target host.

6 Click Next to perform further uninstallation related tasks.**7 A summary window appears. Please follow the instructions presented in this window.**

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

Note – On some Windows systems such as Windows 2000 Advanced Server, the Local Settings folder is a hidden folder. To view this folder and the Temp subdirectory:

Open your Windows Explorer and select Tools → Folder Options from the menu bar. When the Folder Options dialog box is displayed, select the View tab and enable the Show Hidden Files option.

8 Click Close to exit the program.**9 If there are no other connectors installed on the target host, then you can safely remove the *isw-hostname* folder.****10 Repeat “Uninstalling Connectors” on page 162 for all hosts where connectors are installed.**

▼ To Uninstall Core

Note – You must uninstall the Directory Server Plug-in before you uninstall Core.

Uninstalling Core before the Plug-in removes the Plug-in bits without unregistering them from the Directory Server, which will prevent the Directory Server from starting unless you manually remove `cn=pswsync`, `cn=plugins`, `cn=config`.

Use the following instructions to uninstall Core:

1 Start the uninstaller program:

- **On Windows machines:**
 - a. **Click Start, and then choose** Settings → Control Panel.
 - b. **Double-click** Add/Remove Programs.
 - c. **In the Add/Remove Programs window, select** Identity Synchronization for Windows, **then click** Remove.
- **Execute** `runUninstaller.sh` **on Solaris**, `uninstaller.sh` **on Linux**, **or** `uninstall.cmd` **on Windows**.

These programs are located in the installation directory (which is the `/opt/SUNWiw` directory on Solaris and `/opt/sun/isw` directory on Linux by default).

2 In the Welcome screen click Next.

3 Enter the Configuration Directory Host name and Port number.

- a. **Select the root suffix of the configuration directory.** (If necessary, click Refresh to see the list of suffixes.)
- b. **For secure communication between the uninstall program and the configuration directory server, enable the Secure Port box and specify the Directory Server's SSL port number.**

4 Enter your administrator's name and password for the configuration directory.

5 Select Core to be uninstalled and click Next.

6 Enter the configuration directory URL, click Refresh, and select the appropriate root suffix from the drop-down list.

- 7 Click Next to perform further uninstallation related tasks.
- 8 A summary window appears. Please follow the instructions presented in this window.
 - a. On Solaris systems: Uninstallation logs are written to `/var/sadm/install/logs/`
 - b. On Linux systems: Uninstallation logs are written to `/var/sadm/install/logs/`
 - c. On Windows systems: Uninstallation logs are written to the `%TEMP%` directory, which is a subdirectory of the Local Settings folder located under
`C:\Documents and Settings\Administrator`

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

To view this folder and the Temp subdirectory:

Open your Windows Explorer and select Tools → Folder Options from the menu bar. When the Folder Options dialog box is displayed, select the View tab and enable the Show Hidden Files option.

- 9 Click Close to exit the program.

Note – If you are unable to run the connector uninstaller for a given connector for any reason (for example, if you lost the connector files during a hard drive failure), use the `idsync resetconn` subcommand (see [“Using resetconn” on page 215](#)).

This command resets the connector state in the configuration directory to *uninstalled* so that you can reinstall it elsewhere. The `resetconn` subcommand is similar to other commands that access the configuration directory, and it provides two options:

- **-e *dir-source***: Specifies the name of the directory source to be reset. (Connectors are identified in the installers by their directory source name.)
- **-n (safe mode)**: Indicates whether the arguments specified for the command are correct without doing any work.

Example command:

```
idsync resetconn -D "cn=Directory Manager"-w [-h CR-hostname]
[-p 389] [-s dc=example,dc=sun,dc=com] -q [-Z] [-P "cert8.db"]
[-m "secmod.db"] -e "dc=central, dc=example,dc=com" [-n]
```

`resetconn` Output:

```
NOTICE: This program will reset the installation state
to UNINSTALLED for the Connector associated with the
specified DirectorySource 'dc=central,dc=example,dc=com'.
Changing the Connector to an UNINSTALLED state is a
last resort. This is NOT meant to be used for
uninstalling connectors.It is typically used if you
lost a machine with the connector on it and can not
run the uninstaller. Additionally, this program will
rewrite the existing configuration. This can be a
lengthy process. Before proceeding, you should stop
the Console, any running installers, and all other
system processes. You may want to export the ou=Services
tree in the configuration directory to ldif as a backup.
Do you want to reset the installer settings
for the connector (y/n)?
```

Uninstalling the Console Manually

After you have removed all other Identity Synchronization for Windows components, you may have to manually uninstall the Console.

From Solaris or Linux Systems

▼ To Uninstall the Console from Solaris or Linux

- 1 Delete the following subtree from the configuration directory:

```
cn=Sun Java (TM) System Identity Synchronization for Windows,  
cn=server_group,cn=hostname,  
ou=domain_name, o=netscaperoot
```

- 2 For all console installations, remove all of the `.jar` files with an *isw* prefix from the following directory:

```
serverroot/server/java/jars
```

From Windows Systems

▼ To Uninstall the Console from a Windows Active Directory or NT system

- 1 Delete the following subtree from the configuration directory:

```
cn=Sun Java (TM) System Identity Synchronization for Windows,  
cn=server_group, cn=hostname,  
ou=domain_name, o=netscaperoot
```

- 2 For all console installations, remove all of the `.jar` files with an *isw* prefix from the following directory:

```
serverroot/server/java/jars
```


Configuring Security

This chapter provides important information about configuring security for your deployment. The information is organized as follows:

- “Security Overview” on page 169
- “Hardening Your Security” on page 175
- “Securing Replicated Configurations” on page 178
- “Using idsync certinfo” on page 180
- “Enabling SSL in Directory Server” on page 181
- “Enabling SSL in the Active Directory Connector” on page 183
- “Adding Active Directory Certificates to Directory Server” on page 187
- “Adding Directory Server Certificates to the Directory Server Connector” on page 188

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:

- “Specifying a Configuration Password” on page 170
- “Using SSL” on page 170
- “Generated 3DES Keys” on page 171
- “SSL and 3DES Keys Protection Summary” on page 171

- [“Message Queue Access Controls” on page 173](#)
- [“Directory Credentials” on page 174](#)
- [“Persistent Storage Protection Summary” on page 174](#)

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 175](#).

Using SSL

You can configure Identity Synchronization for Windows to use LDAP over SSL everywhere that components use LDAP. All access to Message Queue is protected with SSL.

You must use SSL between the Active Directory Connector and Active Directory when you are synchronizing from Directory Server to Active Directory.

Requiring Trusted SSL Certificates

By default, connectors configured to use SSL will accept any SSL certificate that the server (i.e. Directory Server or Active Directory) returns — which includes untrusted, expired, and invalid certificates. All network traffic between the connector and server will be encrypted, but the connector will not detect a server that is impersonating the true Active Directory or Directory Server.

To force the connector to accept only trusted certificates, use the Console to enable the Require trusted SSL certificates option on the Specify Advanced Security Options panel of the Directory Source Configuration wizard (see [“Creating an Active Directory Source” on page 95](#)). After enabling this option, you must add the appropriate CA certificates to the connector’s certificate database as reported by `idsync certinfo`.

Generated 3DES Keys

A 3DES key generated from the configuration password is used to secure all sensitive information in the product’s configuration directory. With the exception of log messages, all messages to the Message Queue are encrypted with per-topic 3DES keys. Messages sent between connectors and subcomponents are encrypted with per session 3DES keys. The Directory Server Plug-in encrypts all user password changes with a 3DES key.

SSL and 3DES Keys Protection Summary

[“SSL and 3DES Keys Protection Summary” on page 171](#) summarizes how Identity Synchronization for Windows protects sensitive information that is sent over the network.

TABLE 8-1 Protecting Sensitive Information Using Network Security

Use this Protection Method	Between the Following Information Types:
LDAP over SSL (optional)	<ul style="list-style-type: none">■ Directory Server Connector and Directory Server, Active Directory Connector and Active Directory■ Directory Server Plug-in and Active Directory■ Command line interfaces and the product’s configuration directory■ Console and the product’s configuration directory■ Console and Active Directory Global Catalog■ Console and Active Directory domains or Directory Servers being synchronized■ Message Queue broker and the product’s configuration directory■ Connectors, system manager, central logger, command line interfaces, and Console may authenticate the Message Queue over LDAPS■ Installer and the Configuration Directory Server■ Installer and Active Directory■ Installer and the Directory Server being synchronized
Encrypted with 3DES keys (default)	<ul style="list-style-type: none">■ Directory Server Connector and Directory Server Plug-in (all data)■ Windows NT Connector, Windows NT Password Filter DLL, and Windows NT Change Detector (all data)■ All sensitive information in the product’s configuration directory■ All messages sent between connectors and subcomponents (encrypted with per-session 3DES keys)■ All (non-log) messages sent over Message Queue

“[SSL and 3DES Keys Protection Summary](#)” on page 171 contains an overview of the security features discussed in this section.

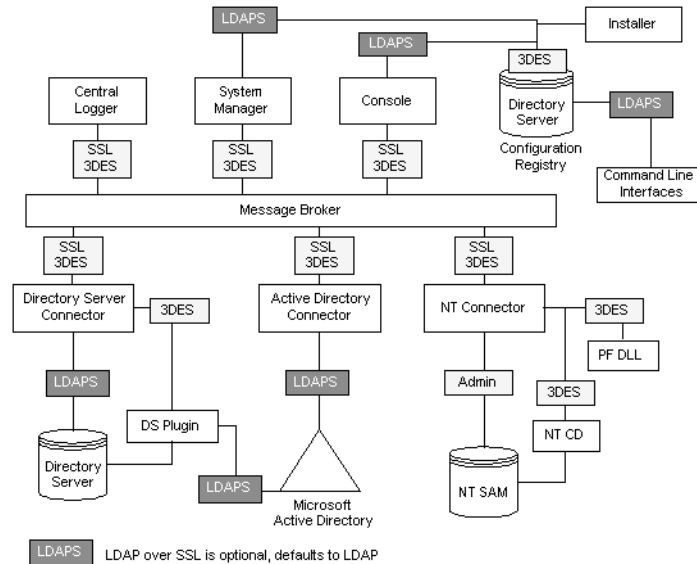


FIGURE 8-1 Security Overview for Identity Synchronization for Windows

Message Queue Access Controls

Identity Synchronization for Windows uses Message Queue's access control to prevent unauthorized access to message subscription and publishing, allowing each connector to trust messages that it receives.

Unique username and passwords known only to Message Queue and to the connector are provided to access the Message Queue broker. Each message sent over the Message Queue is encrypted with a per topic 3DES key, which protects the message contents and prevents outsiders who do not know the topic key from sending meaningful messages. These measures prevent (a) an attacker from sending falsified password synchronization messages to connectors and (b) an attacker from impersonating a connector and receiving actual password updates.

Note – By default, clients of the Message Queue, such as the connectors and system manager, accept any SSL certificate that the Message Queue broker returns. See [“Hardening Your Security” on page 175](#) for more information to enhance Message Queue certificate validation and other Message Queue-related security issues.

Directory Credentials

Privileged credentials are required by the connectors to change passwords in Active Directory and the Directory Servers being synchronized. These privileged credentials are encrypted before they are stored in the product’s configuration directory.

Persistent Storage Protection Summary

“[Persistent Storage Protection Summary](#)” on [page 174](#) summarize how Identity Synchronization for Windows protects sensitive information that is stored on disk.

TABLE 8-2 Persistent Storage Protection

Persistent Storage	Confidential Information	Protection
Product’s Configuration Stored in a Configuration Directory Server	Credentials for accessing the directories and per Message Queue topic 3DES keys are stored in the product’s configuration directory.	All sensitive information stored in the product’s configuration directory is encrypted with a 3DES key that is generated from the configuration password. See “ Hardening Your Security ” on page 175 for recommendations to further protect the product’s configuration directory.
Directory Server Retro Changelog	The Directory Server Plug-in captures password changes and encrypts them before writing them to the Directory Server Retro Changelog.	The Directory Server Plug-in encrypts all user password changes with a 3DES key that is unique to each deployment.
Message Queue Broker Persistent Storage	The Message Queue broker stores password synchronization messages sent between all connectors.	With the exception of log messages, all persisted messages are encrypted with per-topic 3DES keys.
Message Queue Broker Directory Credentials	The Message Queue broker authenticates users against the product’s configuration directory. It connects to the configuration directory using the directory administrative user name and password provided during Core installation.	The directory password is stored in a passfile, which is protected with file system access controls.
System Manager Boot File	The system manager’s boot file contains information for accessing the configuration. This includes the configuration password and the directory administrative user name and password provided during Core installation.	This file is protected with file system access controls.
Connectors and Central Logger Boot Files	Each connector as well as the central logger have an initial configuration file with credentials for accessing the Message Queue.	These files are protected with file system access controls.

TABLE 8-2 Persistent Storage Protection (Continued)

Persistent Storage	Confidential Information	Protection
Directory Server Plug-in Boot Configuration	The Plug-in's configuration, stored in <code>cn=config</code> , includes credentials for connecting to the connector.	The <code>cn=config</code> subtree is protected with AClS and the <code>dse.ldif</code> file, which mirrors this tree, is protected with file system access controls.
NT Password Filter DLL and NT Change Detector Boot Configuration	The NT subcomponent's configuration, which is stored in the Windows registry, includes credentials for connecting to the connector.	If access to the PDCs registry is not secure, these registry keys can be protected with access controls.
Windows Connector's Object Cache	Windows connectors store hashed user passwords in the connector's object cache.	The passwords are not stored in the clear but encrypted with MD5 hashes. These database files are protected with file system access controls. (see "Hardening Your Security" on page 175)

Hardening Your Security

This section depicts potential security weaknesses in the current release of the product and recommendations as to how to extend and harden security outside the product's default configuration. It includes the following:

- "Configuration Password" on page 175
- "Creating Configuration Directory Credentials" on page 175
- "Message Queue Client Certificate Validation" on page 176
- "Message Queue Self-Signed SSL Certificate" on page 177
- "Access to the Message Queue Broker" on page 177
- "Configuration Directory Certificate Validation" on page 177
- "Restricting Access to the Configuration Directory" on page 177

Configuration Password

The configuration password is used to protect sensitive configuration information but the installation program does not enforce any password policy for this password; be sure that this password follows some strict guidelines choose a complex password that is not easily guessed and follow standard policy guidelines for strong passwords.

For example, it should be at least eight characters long, include upper case letters, lower case letters, and non-alphanumeric characters. It should not include your name, initials, or dates.

Creating Configuration Directory Credentials

To access the Directory Server where the product's configuration directory resides, your credentials must be in the Configuration Administrators group. However, if you need to create credentials other than *admin* for any reason, consider the following:

The installation program requires you to provide credentials for a user stored in the Console administrative subtree. However, the Core installation program will not expand users other than *admin* into “uid=admin,ou=Administrators, ou=TopologyManagement, o=NetscapeRoot”. Therefore, you must specify the entire DN during Core installation.

▼ To Create a New User Other Than *admin*

1 Create a user in:

ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

2 Add the new credentials to the Configuration Administrators group.

3 Set ACIs to allow only this user or all users in the Configuration Administrators group to access the Directory Server where the product's configuration directory is stored.

4 Specify entire DN during Core installation.

For more information about managing access controls in the Directory Server, see [Chapter 6, “Directory Server Access Control,” in *Sun Directory Server Enterprise Edition 7.0 Administration Guide*](#)

Message Queue Client Certificate Validation

By default, clients of the Message Queue, such as the connectors and system manager, accept any SSL certificate that the Message Queue broker returns.

▼ To Validate the Message Queue Client Certificate

1 To override this setting and force Message Queue clients to validate the Message Queue broker's certificate, edit:

installation_root/resources/WatchList.properties

2 Add the following to the JVM arguments of each process in *Watchlist.properties*:

`-Djavax.net.ssl.trustStore=keystore_path-DimqSSLIsHostTrusted=false`

3 Restart the Identity Synchronization for Windows daemon or service.

The `javax.net.ssl.trustStore` property should point to a JSEE keystore that trusts the broker certificate, for example, `/etc/imq/keystore` can be used on the machine where Core was installed because this is the same keystore used by the broker.

Message Queue Self-Signed SSL Certificate

By default, the Message Queue broker uses a self-signed SSL certificate. To install a different certificate, use the `keytool` utility that ships with Java to modify the broker's keystore (`/var/imq/instances/isw-broker/etc/keystore` on Solaris, `/var/opt/sun/mq/instances/isw-broker/etc/keystore` on Linux, and `mq_installation_root/var/instances/isw-broker/etc/keystore` on Windows 2000). The alias of the certificate must be `imq`.

Access to the Message Queue Broker

By default, the Message Queue uses dynamic ports for all services except for its port mapper. To access the broker through a firewall or restrict the set of hosts that can connect to the broker, the broker should use fixed ports for all services.

This can be achieved by setting the `imq.service_name protocol_type .port` broker configuration properties. Refer to the *Sun Java System Message Queue Administration Guide* for more information.

Configuration Directory Certificate Validation

The system manager accepts any certificate when connecting to the product's configuration directory over SSL; the Message Queue broker accepts any certificate when connecting to the product's configuration directory over SSL. Currently, there is no way to make either the system manager or the Message Queue broker validate the product's configuration directory SSL certificates.

Restricting Access to the Configuration Directory

When Core is installed, the process of adding information to the Directory Server where the product's configuration directory is stored does not include adding any access control information. To restrict access to only configuration Administrators, the following ACI can be used:

```
(targetattr = "*")
(target = "ldap://ou=IdentitySynchronization,
ou=Services,dc=example,dc=com")
(version 3.0;acl "Test";deny (all)
(groupdn != "ldap://cn=Configuration Administrators,
ou=Groups, ou=TopologyManagement, o=NetscapeRoot");)
```

For more information about managing access controls in the Directory Server, see [Chapter 6, “Directory Server Access Control,”](#) in *Sun Directory Server Enterprise Edition 7.0 Administration Guide*

Securing Replicated Configurations

Deployments connecting to Directory Servers using replication follow the same rules identified in [“Security Overview” on page 169](#). This section gives an example replicated configuration and explains how to enable use of SSL in this configuration.

Note – For an overview of planning, deploying, and securing replicated configurations see [Appendix D, “Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows”](#)

[“Securing Replicated Configurations” on page 178](#) lists the configuration components requiring CA certificates and identifies which certificates are required where.

TABLE 8-3 MMR Configuration Components Requiring CA Certificates

Component	Required CA certificates
Preferred Directory Server Replicated Master	Active Directory System
Secondary Directory Server Replicated Master	Active Directory System
Read-only Directory Server Hub(s)	Preferred Directory Server Replicated Master
	Secondary Directory Server Replicated Master
Directory Server Connector	Preferred Directory Server Replicated Master
	Secondary Directory Server Replicated Master
Active Directory Connector	Active Directory System

Replicated configuration shows Identity Synchronization for Windows installed in an MMR configuration, where there are two replicated Directory Server masters with multiple Directory Server read-only hubs or consumers. Each Directory Server has a Plug-in and there is only one Directory Server Connector, one Active Directory system, and one Active Directory Connector.

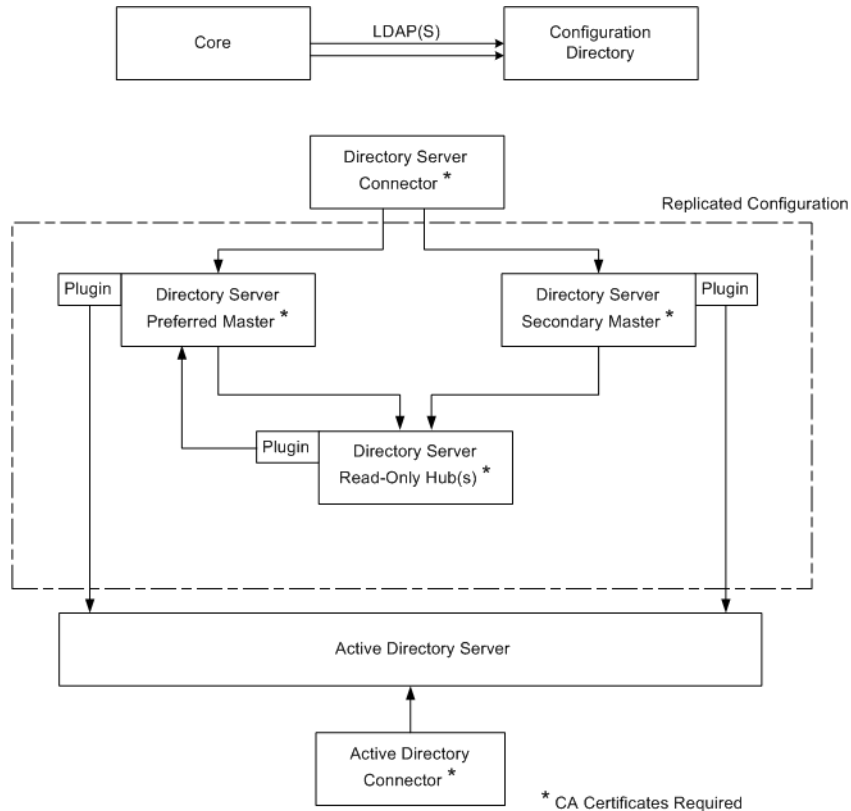


FIGURE 8-2 Replicated Configuration

When the Directory Server source is configured for SSL, you must make sure that both the preferred and secondary Directory Server certificates are trusted by the replica Directory Server. This is true for every Directory Server Plug-in of type other than that you install on a system with a Directory Server hub or read-only replica.

Note – Directory Server Plug-ins have access to the same CA certificates as its associated Directory Server.

The above diagram is specific to two Directory Server masters. But you can extend this to contain multiple masters.

Using idsync certinfo

Use the `idsync certinfo` utility to determine what certificates are required based on the current Identity Synchronization for Windows SSL settings. Execute `idsync certinfo` to retrieve information about what certificates are required in each certificate database.

Note – You must be sure that when you are configuring the Directory Server source for SSL, both the preferred and secondary Directory Server source certificates are trusted by the replica Directory Server for all Directory subcomponents or Plug-ins.

If Identity Synchronization for Windows tries to establish SSL connections (with the trust all certificates setting enabled), and the server's hostname does not match the hostname provided in the certificate presented by the server during the SSL negotiation phase, the Identity Synchronization for Windows Connector will refuse to establish the connection.

The directory source hostname in the Identity Synchronization for Windows configuration must always match the hostname embedded in the certificate used by that directory source.

Arguments

Arguments describes the arguments you can use with the `idsync certinfo` subcommand.

TABLE 8-4 certinfo Arguments

Argument	Description
-h <i>CR-hostname</i>	Specifies the configuration directory hostname. This argument defaults to the values specified during Core installation.
-p <i>CR-port-no</i>	Specifies the configuration directory LDAP port number. (<i>Default is 389</i>)
-D <i>bind-DN</i>	Specifies the configuration directory bind distinguished name (DN). This argument defaults to the values specified during Core installation.
-w <i>bind-password</i> -	Specifies the configuration directory bind password. The - value reads the password from standard input (STDIN).
-s <i>rootsuffix</i>	Specifies the configuration directory rootsuffix. Where rootsuffix is a distinguished name such as <code>dc=example,dc=com</code> . This argument defaults to the values specified during Core installation.
-q <i>configuration_password</i>	Specifies the configuration password. The - value reads the password from standard input (STDIN).

Usage

The following example uses `idsync certinfo` to search for system components designated to run under SSL communications. The results of this example identifies two connectors (CNN101 and CNN100) and provides instructions as to where to import the appropriate CA certificate.

```
: \Program Files\Sun\MPS\isw-
hostname\bin idsync certinfo -h
CR-hostname -p 389 -D
"cn=Directory Manager" -w dirmanager -s dc=example,dc=com
-q password
Connector: CNN101
Certificate Database Location: C:\Program Files\Sun\MPS\isw-
hostname\etc\CNN101
Get 'Active Directory CA' certificate from Active Directory
and import into Active Directory Connector certificate db
for server ldaps://
hostname.example.com:636
Connector: CNN100 Certificate Database Location:
C:\Program Files\Sun\MPS\isw-
hostname\etc\CNN100
Export 'Directory Server CA' certificate
from Directory Server certificate db and
import into Directory Server Connector certificate db
ldaps://hostname.example.com:636
Export 'Active Directory CA' certificate
from Active Directory Server
hostname.example.sun.com:389
and import into Directory Server Server certificate db
for server ldaps://hostname.example.com:638
SUCCESS
```

Enabling SSL in Directory Server

Follow these steps to enable SSL in a Directory Server using a self-signed certificate.

Note – These abbreviated procedures are for your convenience. Refer to the [Sun Directory Server Enterprise Edition 7.0 Administration Guide](#) for more information.

- On Windows, use the certutil version bundled with Identity Synchronization for Windows 6.0 within the *ISW-host-name\shared\bin* folder.
 - On Solaris, certutil is installed in */usr/sfw/bin* by default.
 - On Linux, certutil is installed in */opt/sun/private/bin* by default.
-

▼ To Enable SSL in Directory Server

Refer to the following procedure to enable SSL in Directory Server:

1 Create a DS instance

```
/opt/SUNWdsee/ds6/bin/dsadm create -p non-ldap-port -P ldap-secure-port  
<DS-server-root>/slapd-<hostname>
```

2 Start the instance

```
/opt/SUNWdsee/ds6/bin/dsadm start <DS-server-root>/slapd-<hostname>
```

3 Create a self-signed certificate

```
/opt/SUNWdsee/ds6/bin/dsadm add-selfsign-cert -S "cn=<machine name with  
domain>,O=<preferred root suffix>" /<DS-server-root>/slapd-<hostname>/<certificate name>
```

Where S = Create an individual certificate and add it to database, the second variable represents the path of Directory Server instance and the last variable is for the certificate alias.

4 Set the server properties to this certificate

```
/opt/SUNWdsee/ds6/bin/dsconf set-server-prop -p non-ldap-port  
ssl-rsa-cert-name:<certificate name>
```

5 Restart the DS

```
/opt/SUNWdsee/ds6/bin/dsadm restart /<DS-server-root>/slapd-<hostname>/
```

6 Now stop the DS and remove the default certificate (this ensures that the above generated certificate will be the default certificate)

```
/opt/SUNWdsee/ds6/bin/dsadm stop /<DS-server-root>/slapd-<hostname>/
```

7 Now remove the default certificate

```
/opt/SUNWdsee/ds6/bin/dsadm remove-cert /<DS-server-root>/slapd-<hostname>/  
defaultCert
```

where the first variable represents the slapd-path and the second variable represents the alias of the certificate. In case you want to export the above default certificate, following is the command

```
/opt/SUNWdsee/ds6/bin/dsadm export-cert -o /<any path>/slapd-cert.export  
/<DS-server-root>/slapd-<hostname>/ <original default cert alias>
```

where o=output file (/<any path>/slapd-cert.export), the second variable represents the slapd-path and the third variable represents the certificate alias.

Retrieving the CA Certificate from the Directory Server Certificate Database

Ensure that you have enabled SSL in Directory Server. To export the Directory Server certificate to a temporary file so that you can import it into the certificate database of the Directory Server Connector, issue the following command:

```
<ISW-server-root>\shared\bin\certutil.exe -L -d .  
-P slapd-hostname- -n server-cert -a \ > C:\s-cert.txt
```

ISW-server-root is the path where ISW-hostname directory is present.

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

Retrieving the CA Certificate from the Directory Server (using dsadm command on Solaris platform)

Ensure that you have enabled SSL in Directory Server. To retrieve the CA certificate issue the following command:

```
/opt/SUNWdsee/ds6/bin/dsadm export-cert -o /<any path>  
/slapd-cert.export /<DS-server-root>/slapd-<hostname>/  
<original default cert alias>
```

Enabling SSL in the Active Directory Connector

Identity Synchronization for Windows *automatically* retrieves Active Directory SSL certificates over SSL and imports them into the Connector's certificate database using the same credentials you provided for the Connector.

However; if an error occurs (for example, invalid credentials or no SSL certificates were found), you can retrieve an Active Directory CA certificate and add it to the Connector certificate database. See the following sections for instructions:

- [“Retrieving an Active Directory Certificate” on page 184](#)
- [“Adding Active Directory Certificates to the Connector’s Certificate Database” on page 185](#)

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+Tu14NGoQnxhmNHLjANBgk
qhkiG9w0BAQUFADCBjjEeMBwGCSqGSIb3DQEJARYPYmVydG
9sZEBzdW4uY29tMQswCQYDVQQGEwJVUzELMAkGA1UECBMCV
FgxDzANBgNVBACtBkF1c3RpbjEzMBCGA1UEChMQU3VvIE1p
Y3Jvc3lzdGVtczEQMA4GA1UECXMHaVBsYW5ldDEUMBIGA1U
EAXMLUmVzdGF1cmFudHMwHhcNMDIwMTEwMTEwMDA1NDA5WhcNMT
IwMTEwMDE1OTQ2WjCBjjEeMBwGCSqGSIb3DQEJARYPYmVyd
G9sZEBzdW4uY29tMQswCQYDVQQGEwJVUELMAkGA1UECBMCV
FgxDzANBgNVBACtBkF1c3RpbjEzMBCGA1UEChMQU3VvIE1p
Y3Jvc3lzdGVtczEQMA4GA1UECXMHaVBsYW5ldDEUMBIGA1U
EAXMLUmVzdGF1cmFudHMwXDNBkgqhkiG9w0BAQEFAANLAD
BIAkEAyekZa8gwwhw3rLK3eV/12St1DVUsg31LOu3CnB8cM
HQZXLgiUgtQ0hm2kpZ4nEhwCAHhFLD3iIhIP4BGWQFjcwID
AQABo4IBnjCCAZowEwYJKwYBBAGCNxQCBAYeBABDAEEwCwY
DVR0PBAQDAgFGMA8GA1UdEwEB/wQFMAMBAf8wHQYDVR0OBB
YEFJ5Bgt60ypq7T80ykw4LH6ws2d/IMIIBMyDVR0fBIIBK
TCCASUwgdOggdGgc2GgcpsZGFw0i8vL0NOPVJlc3RhdXJh
bnRzLENOPWRvd2l0Y2hlcixDTj1DRFAsQ049UHVibGljJTI
wS2V5JTlWU2VydmljZXMsQ049U2VydmljZXMsQ049Q29uZm
lndXJhdGlvbixEQz1yZXN0YXVyYW50cyxEQz1jZW50cmFsL
RPXN1bixEQz1jb20/Y2VydGlmaWNhdGVScXZvY2F0aW9u
TGlzdD9iYXNlP29iamVjdGNSYXNzPWNSTERpc3RyaWJ1dG1
vb1BvaW50ME2gS6BJhkdodHRwOi8vZG93aXRjaGVyLnJlc3
RhdXJhbnRzLmNlbnRyYWwuc3VuLmNvbS9DZXJ0RW5yb2xsL
1Jlc3RhdXJhbnRzLmNybDAQBgkrBgEEAYI3FQEEAwIBADAN
BgkqhkiG9w0BAQUFAANBAL5R9R+ONdDVHwu/5Sd9Tn9dpXN
8oegjS88ztv1HD6XSTDzGTuaaVebSVZ3I+ghSInsgQbH0gW
4fGRwaI BvePI4=
-----END CERTIFICATE-----
```

- 4 **Save the certificate into a file (such as ad-cert.txt).**
- 5 **You can then import that file (for example, ad-cert.txt) into a certificate database. Continue to the next section, [“Adding Active Directory Certificates to the Connector’s Certificate Database” on page 185](#)**

Adding Active Directory Certificates to the Connector’s Certificate Database

Use this procedure only if you enabled SSL for the Active Directory Connector after installing the Connector or if invalid credentials were provided during installation.

▼ To Add Active Directory Certificate to the Connector's Certificate Database

- 1 On the machine where the Active Directory Connector is installed, stop the Identity Synchronization for Windows service/daemon.
- 2 Retrieve the Active Directory CA certificate using one of the following methods:
 - “Using Window’s Certutil” on page 184
 - “Using LDAP” on page 184
- 3 Assuming the Active Directory Connector has connector ID CNN101 (see `logs/central/error.log` for a mapping from connector ID to the directory source it manages), go to its certificate database directory on the machine where it was installed, and import the certificate file:

- If the certificate was retrieved using `certutil`, type:

```
<ISW-server-root>\shared\bin\certutil.exe -A -d . -n ad-ca-cert -t C,, -i \cacert.bin
```

- If the certificate was retrieved using LDAP, type:

```
<ISW-server-root>\shared\bin\certutil.exe -A -d . -n ad-ca-cert \
-t C,, -a -i \ad-cert.txt
```

ISW-server-root is the path where ISW-hostname directory is present

On Solaris, the certificate can be imported using the `dsadm` command in the following manner:

```
/opt/SUNWdsee/ds6/bin/dsadm add-cert -C <DS-server-root>/slapd-<hostname>/ ad-ca-cert cacert.bin
```

where `ad-ca-cert` is the name of the certificate assigned after the import and `cacert.bin` is the certificate about to be imported

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

Note – Because the Directory Server `certutil.exe` is installed automatically when you install Directory Server, 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 package on the server where the Active Directory Connector is installed. (You do not have to install the Administration Server or Directory Server components.)

In addition, be sure to select the JRE subcomponent from the Console (to ensure your ability to uninstall).

Adding Active Directory Certificates to Directory Server

Note – Make sure that you have enabled SSL in Directory Server.

▼ To Add the Active Directory CA certificate to the Directory Server Certificate Database

- 1 Retrieve the Active Directory CA certificate using one of the following methods:
 - [“Using Window’s Certutil” on page 184](#)
 - [“Using LDAP” on page 184](#)
- 2 Stop Directory Server.
- 3 Import `cacert.bin` into the `<DS-server-root>\slapd-hostname\alias` folder on Windows and for Solaris and Linux import it into `<DS-server-root>/slapd-hostname/alias` directory.
- 4 On the machine where Directory Server is installed, import the Active Directory CA certificate as follows:

- If the certificate was retrieved using `certutil`, type:

```
<ISW_server_root>\shared\bin\certutil.exe -A -d .  
-P slapd-hostname- -n ad-ca-cert -t C,, -i \cacert.bin
```

- If the certificate was retrieved using LDAP, type:

```
<ISW_server_root>\shared\bin\certutil.exe -A -d .  
-P slapd-hostname- -n ad-ca-cert -t C,, -a -i \ad-cert.txt
```

ISW-server-root is the path where ISW-hostname directory is present

- If the certificate was retrieved using the dsadm command (on Solaris), type:

```
/opt/SUNWdsee/ds6/bin/dsadm add-cert -C <DS-server-root>  
/slapd-<hostname>/ ad-ca-cert cacert.bin
```

Where ad-ca-cert is the name of the certificate assigned after the import and cacert.bin is the certificate about to be imported

5 Start Directory Server.

Adding Directory Server Certificates to the Directory Server Connector

If you enable SSL communication between the Directory Server Plug-in and Active Directory, then you must add the Active Directory CA Certificate to the certificate database of each Directory Server master.

▼ To Add the Directory Server Certificates to the Directory Server Connector

- 1 On the machine where the Directory Server Connector is installed, stop the Identity Synchronization for Windows service/daemon.
- 2 Retrieve the Directory Server CA certificate.
- 3 Assuming the Directory Server Connector has connector ID CNN100 (see logs/example/error.log for a mapping from connector ID to the directory source it manages), go to its certificate database directory on the machine where it was installed, and import the cacert.bin file:

```
<ISW_server_root>\shared\bin\certutil.exe -A -d . -n ds-cert -t C,, -i C:\s-cert
```

ISW-server-root is the path where ISW-hostname directory is present.
- 4 Restart the Identity Synchronization for Windows service/daemon.

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:

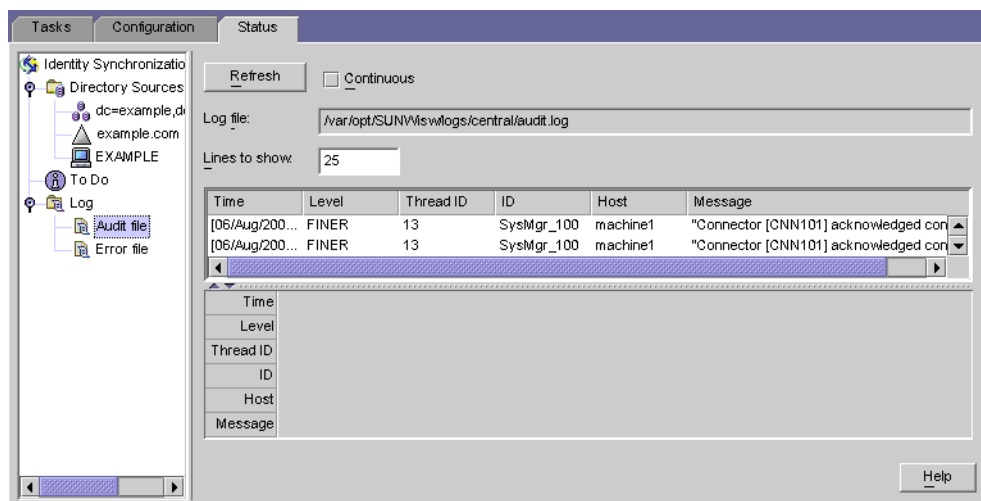
- [“Understanding the Logs” on page 189](#)
- [“Configuring Your Log Files” on page 194](#)
- [“Viewing Directory Source Status” on page 196](#)
- [“Viewing Installation and Configuration Status” on page 197](#)
- [“Viewing Audit and Error Logs” on page 198](#)
- [“Enabling Auditing on a Windows NT Machine” on page 199](#)

Understanding the Logs

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

If you select one of the following nodes in the navigation tree pane (on the left), the content presented on the Status tab changes to provide specific information about that item.

- **Directory Source:** Select a directory source node (such as `dc=example,dc=com`) to view status information about that directory source.
- **To Do:** Select this node for a list of the steps you must complete to successfully install and configure Identity Synchronization for Windows (the program greys-out all completed steps).
- **Audit File:** Select this node for information about day-to-day system operations (including error conditions).
- **Error File:** Select this node for information about error conditions on your system. (The Error log essentially acts as a filter in which only the error entries are displayed.)



Log Types

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

- “Central Logs” on page 190
- “Local Component Logs” on page 191
- “Local Windows NT Subcomponent Logs” on page 192
- “Directory Server Plug-in Logs” on page 192

Central Logs

As long as Identity Synchronization for Windows components can access Message Queue, all audit and error messages will be logged in the Identity Synchronization for Windows central logger. Consequently, these central logs (which include messages from all components) are the primary logs to monitor.

The centralized logs are located on the machine where Core is installed, in the following directories:

- **On Solaris:** `/var/opt/SUNWisw/logs`
- **On Linux:** `/var/opt/sun/isw/logs`
- **On Windows:** `installation_root\isw-machine_name\logs\central\`

TABLE 9-1 Log Types for Identity Synchronization for Windows

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
[example.com (ldaps:// server1.example.com:636)];
CNN101 is the connector that manages
[dc=example,dc=com (ldap:// server2.example.com:389)];"
```

In addition to the central logger, each component has its 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 9-2 Local Logs

Log Name	Description
audit.log	A superset of error.log that includes messages about each synchronization event. These messages are also written to the central audit.log.
error.log	Warning and Severe messages are reported here. These messages are also written to the central error.log.

These local logs are located in the following subdirectories:

- **On Solaris:** /var/opt/SUNWisw/logs
- **On Linux:** /var/opt/sun/isw/logs
- **On Windows:** *installation_root/isw-machine_name/logs/central/*

The sysmgr and clogger100 (central logger) directories are on the machine where Core is installed.

Identity Synchronization for Windows rotates these local component logs daily by moving the current log to a log file that includes the date, as follows:

```
audit_2004_08_06.log
```

Note – By default, Identity Synchronization for Windows deletes connector logs after ten days. You can extend this period by editing the `com.sun.directory.wps.logging.maxmiumDaysToKeepOldLogs` value in the `Log.properties` file and restarting the service daemon.

Local Windows NT Subcomponent Logs

The following Windows NT subcomponents also have local logs:

- Windows NT Change Detector DLL
- Password Filter DLL

These subcomponent logs are located in the SUBC1XX (for example, SUBC100) subdirectories of the following directory:

```
installation_root/isw-machine_name/logs/
```

Identity Synchronization for Windows limits these files to 1 MB in size, and keeps only the last 10 logs.

Directory Server Plug-in Logs

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

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

By default, only minimal Plug-in log messages are displayed in the error log. For example:

```
[14/Jun/2004:17:08:36 -0500] - ERROR<38747> - isw - conn=-1  
op=-1 msgId=-1 - Plug-ins unable to establish connection to DS Connector  
at attila:1388, will retry later
```

▼ To Change the Verbosity Level of the Error Logs

You can change the default verbosity level of the Directory Server error log through DSCC as follows:

- 1 **Log in to Directory Service Control Center.**
- 2 **On the Directory Servers tab page, click the server whose log level you want to configure.**

- 3 Select the **Server Configuration** tab, then the **Error Logging** tab.
- 4 In the **General** → **Additional Items to Log** section, select **Plug-Ins**.
- 5 Click **Save**.

You can enable plug-in logging using the command line.

```
$ dsconf set-log-prop errors level:err-plugins
```

For more information about Directory Server logging, refer to [Chapter 14, “Directory Server Logging,”](#) in *Sun Directory Server Enterprise Edition 7.0 Administration Guide*.

Reading the Logs

Every log message includes the following information:

- **Time:** Indicates when (time and date) the log entry was generated. For example:

```
[13/Aug/2004:06:14:36:753 -0500]
```

- **Level:** Indicates the severity and verbosity of the log message. Identity Synchronization for Windows uses the following log levels:

TABLE 9-3 Log Levels

Log Level	Description
INFO	These messages provide a minimum amount of information about each action so you can see that the system is running correctly. For example, you can see when a change is detected and when synchronization occurs. These messages are always logged to the audit log.
FINE	These messages contain more information about an action as it travels through the system.
FINER	These messages contain even more information about an action as it travels through the system. Turning the logging level to FINER for all components may impact performance.
FINEST	These messages contain the most information about an action as it travels through the system. Turning the logging level to FINEST for all components may significantly impact performance.

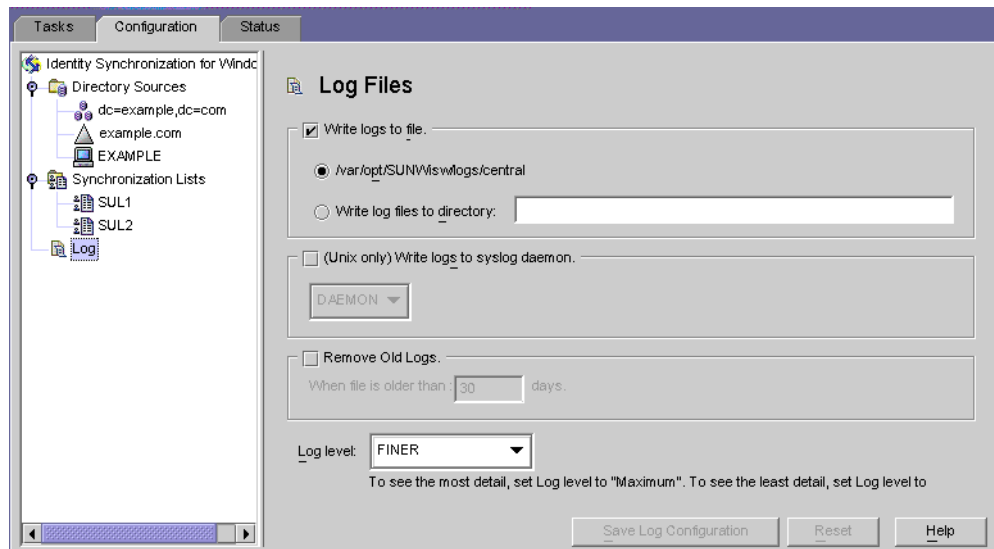
- **Thread ID:** Displays the Java thread ID of the function causing the event.
- **ID:** Identifies the component (console, system manager, and so forth.) causing the event.
- **Host:** Displays the name of the host causing the event.
- **Message:** Displays audit or error information associated with the event. Some examples include:

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

Configuring Your Log Files

▼ To Configure Logging for Your Deployment

- 1 Open the Console and select the Configuration tab.
- 2 In the navigation tree pane, and expand nodes until you see the Logs node.
- 3 Select the Logs node and the Log Files panel is displayed on the Configuration tab.



- 4 Use the Log Files pane to configure your log files, as follows:
 - **Write logs to file.** Enable this option to write logs to a file on the Core host.
 After selecting this option you can:
 - Enable the default log directory and file (for example, /var/opt/SUNWlswwlogs/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 semicolon separated list.

- **Remove Old Logs**: The number of log files will continue to grow (one per day) indefinitely. To avoid running out of disk space, enable this option and specify when the program can delete old logs from the central log file.
For example, if you specify 30 days, Identity Synchronization for Windows will delete all files when they become 31 days old.
- **Log Level**. Use the drop-down list to select the level of detail you want to see in your system logs. (Review [“Reading the Logs” on page 193](#))

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

Viewing Directory Source Status

▼ To View the Status of your Directory Sources

- 1 From the Identity Synchronization for Windows Console, select the Status tab.
- 2 In the navigation tree pane, expand the Directory Source node, and then select the directory source node (such as dc=example,dc=com).

The Status tab content changes to provide information related to the selected directory source.



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

Click Update to refresh the information on this tab. The following information is provided on the Status tab:

- **State:** Reflects the current state of the directory source. Valid states include:
 - **Uninstalled:** The connector has not been installed.
 - **Installed:** The connector is installed, but is not ready for synchronization because it has not received its runtime configuration yet. If the connector remains in this state for more than a minute, something is probably wrong.
 - **Ready:** The connector is ready for synchronization, but it is currently not synchronizing any objects. A connector remains in the Ready state if synchronization has not been started or if synchronization has been started but not all subcomponents have established connections with the connectors.
 - **Syncing:** The connector is synchronizing objects. There might still be errors, so consult the error log if you notice that changes are not synchronized.
- **Active:** Indicates whether the directory source is active or down.

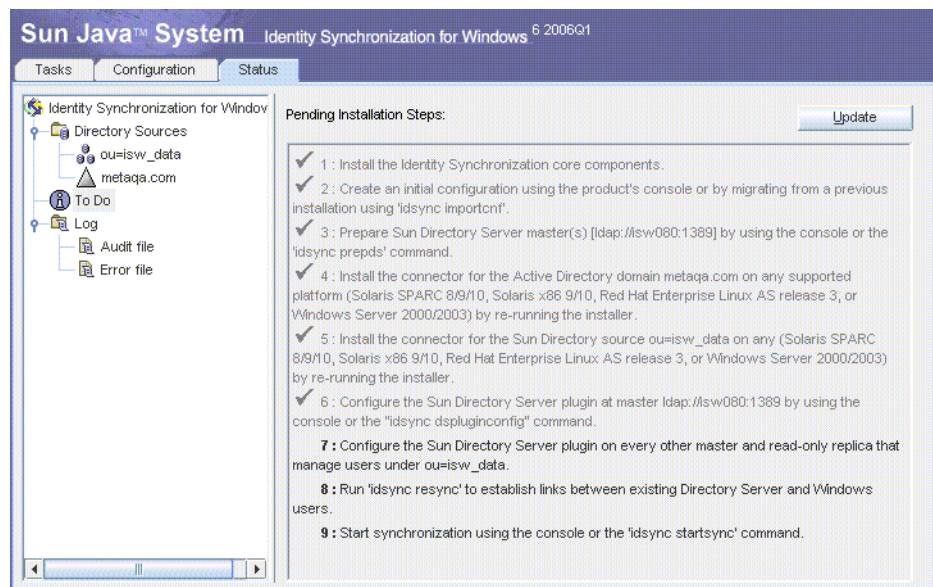
- **Last Communication:** Indicates the time of the last response from this directory source's connector.

Viewing Installation and Configuration Status

▼ To View the Remaining Steps of the Installation and Configuration Process

- 1 From the Identity Synchronization for Windows Console, select the **Status** tab.
- 2 In the navigation tree pane, expand the **To Do** node.

The Status tab content changes to provide a checklist of the installation and configuration steps (for example, see “[Viewing Directory Source Status](#)” on page 196).



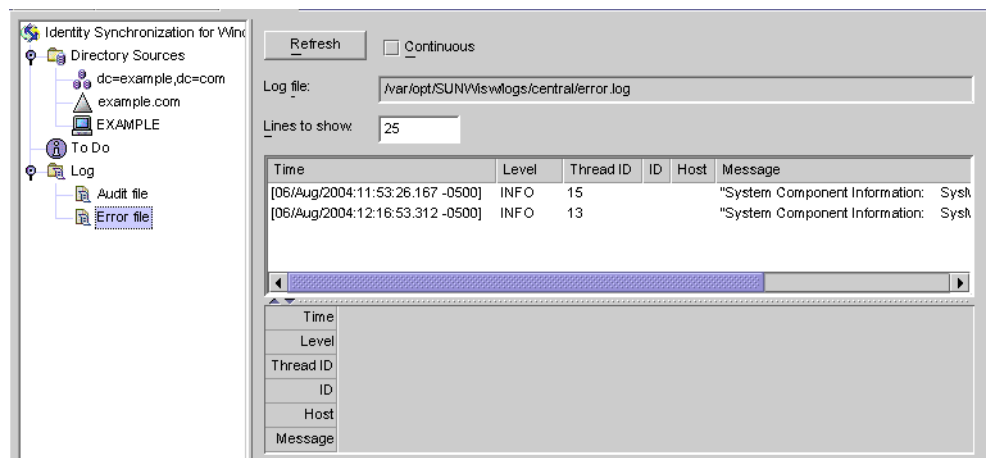
- 3 Click the **Update** button (upper right) to refresh the list.

Completed steps will be check-marked and greyed-out. You must complete the remaining steps to successfully complete the installation and configuration process.

Viewing Audit and Error Logs

▼ To View Your Error Logs

- 1 From the Identity Synchronization for Windows Console, select the Status tab.
 - 2 In the navigation tree pane, expand the Audit File or the Error File node.
- The Status tab content changes to display the current logs.



Click Refresh to load the latest audit or error information.

The following information is provided on the Status tab:

- **Continuous:** Updates and displays the latest audit or error information constantly.
- **Log File:** Displays the full path name of the audit or error log being read; for example:

C:\Program Files\Sun\MPS\isw-*hostname*\logs\central\audit.log
- **Lines to show:** Specifies how many audit or error entries to display. (*Default is 25.*)

Enabling Auditing on a Windows NT Machine

If you have a Windows NT machine in your deployment, verify that auditing is enabled or Identity Synchronization for Windows cannot log messages from that machine.

▼ To Enable Audit Logging on Your Windows NT Machine

- 1 From the Windows NT Start menu, select Programs > Administrative Tools > User Manager for Domains.
- 2 When the User Manager dialog box is displayed, select Policies > Audit from the menu bar. The Audit Policy dialog box is displayed.
- 3 Enable the Audit These Events button and then enable the Success and Failure boxes.
- 4 Click OK to close the dialog box.

These settings will remain in effect until you change them again.

PART II

Identity Synchronization for Windows
Appendixes

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” on page 203](#)
- [“Using the `idsync` command” on page 206](#)
- [“Using the `forcepwchg` Migration Utility” on page 221](#)

Common Features

The Identity Synchronization for Windows command line utilities share the following features:

- [“Common Arguments to the `Idsync` Subcommands” on page 203](#)
- [“Entering Passwords” on page 205](#)
- [“Getting Help” on page 206](#)

Common Arguments to the `Idsync` Subcommands

This section describes the arguments (options) that are common to most of the command line utilities. The information is organized into the following tables:

- **Common Arguments to the `Idsync` Subcommands:** Describes the following arguments, which are common to all of the `idsync` subcommands (*except `prepds`*) and migration tools.

```
-D bind-DN -w bind-password | - [-h Configuration Directory-hostname]
[-p Configuration Directory-port-no] [-s rootsuffix] [-Z] [-P cert-db-path]
[-m secmod-db-path]
```

Note – Brackets [] indicate optional arguments.

The Identity Synchronization for Windows installation program automatically writes default values to the -h, -p, -D, and -s arguments based on the information you provide during installation. However, you can specify a different value on the command line to override a defaulted value.

To support multibyte characters, Identity Synchronization for Windows base64-encodes the default values for -s *rootsuffix* and -D *bind-DN* in the command line interface (CLI) environment file. The rootsuffix default should not be changed. The bind DN default can be overridden on the command line or updated with the appropriate base64-encoded value in the CLI environment file.

- **Common Arguments for Accessing the Configuration Directory Server using SSL:** Describes optional arguments that provide information about securely accessing the Configuration Directory Server using Secure Socket Layer (SSL). These arguments are also common to all of the *idsync* subcommands and the migration tools.
- **Common Arguments Related to Configuration Directory:** Describes arguments related to the configuration directory. These arguments are common to two or more *idsync* subcommands and migration tools.

TABLE A-1 Arguments Common to All Subcommands

Argument	Description
-h <i>Configuration Directory-hostname</i>	Specifies the configuration directory hostname. This argument defaults to the values specified during Core installation.
-p <i>Configuration Directory-port</i>	Specifies the configuration directory LDAP port number.
-D <i>bind-DN</i>	Specifies the configuration directory bind distinguished name (DN). This argument defaults to the values specified during Core installation.
-w <i>bind-password</i> -	Specifies the configuration directory bind password. The - value reads the password from standard input (STDIN).
-s <i>rootsuffix</i>	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 <i>configuration_password</i> -	Specifies the configuration password. The - value means the password will be read from standard input (STDIN). This argument is <i>mandatory</i> for all subcommands except <i>prepds</i> .

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 <i>cert-db-path</i>	<p>Specifies the path and file name of the client's certificate database.</p> <p>This certificate database must contain the CA certificate used to sign the Directory Server's certificate database.</p> <p>If you specify -Z but do not use -P, the <i>cert-db-path</i> defaults to <i>current-working-directory/cert8.db</i>.</p> <p>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: <i>cert8.db</i>, <i>key3.db</i>, and <i>secmod.db</i>.</p>
-m <i>secmod-db-path</i>	<p>Specifies the path to the security module database. For example:</p> <p><i>/var/Sun/MPS/slaped-serverID/secmod.db</i></p> <p>Specify this argument only if the security module database is in a different directory than the certificate database itself.</p>

TABLE A-3 Configuration Directory Arguments

Argument	Description
-a <i>ldap_filter</i> Use with <i>forcepwchg</i> and <i>resync</i> 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 <i>filename</i> Use with <i>export10cnf</i> , <i>importcnf</i> , and <i>resync</i> subcommands	Specifies the name of a Configuration XML Document file.
-n Use with <i>forcepwchg</i> , <i>importcnf</i> , and <i>resetconn</i> subcommands	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Entering Passwords

Wherever a password argument is required (such as *-w bind-password* or *-q configuration_password*), you can use the “-” argument to tell the password program to read the password from STDIN.

If you use the “-” value for multiple password options, `idsync` will prompt you for passwords based on the arguments’ order.

In this case, the program would expect the *bind-password* first, and then for the *configuration-password*.

Getting Help

You can use one of the following commands to display usage information about `idsync` or any of its subcommands in the command Console:

- **-help**
- **--help**
- **-?**

For usage information

- About `idsync` (including a list of valid subcommands), type one of the preceding help options at a command prompt and click Return.
- About a subcommand, type the subcommand followed by a help option at a command prompt and click Return.

Using the `idsync` command

You use the `idsync` command and subcommands to execute the Identity Synchronization for Windows command line utility.

Note – The `idsync` command converts all DN-valued arguments (such as bind DN or suffix name) from the character set specified for that window to UTF-8 before sending the arguments to Directory Server.

Do not use backslashes as escape characters in suffix names.

To specify UTF-8 characters on Solaris and on Linux, your terminal window must have a locale based on UTF-8. Make sure that the environmental variable’s `LC_CTYPE` and `LANG` are set correctly.

Unless specifically noted otherwise, you can run the `idsync` command with subcommands using either of the following methods:

- **From Solaris:**
 1. Open a terminal window and **cd** to the `/opt/SUNWsw/bin` directory.
 2. Type the `idsync` command with one subcommand, as follows

idsync subcommand

■ **From Linux:**

1. Open a terminal window and **cd** to the `/opt/sun/isw/bin` directory.
2. Type the `idsync` command with one subcommand, as follows

idsync *subcommand*

■ **From Windows:**

1. Open a Command Window and **cd** to the `install_path\isw-hostname\bin` directory.
2. Type the `idsync` command with one subcommand, as follows

idsync *subcommand*

“Using the `idsync` command” on page 206 lists all of the `idsync` utility subcommands and their purpose:

TABLE A-4 Quick Reference to `idsync` Subcommands

Subcommand	Purpose
<code>certinfo</code>	Displays certificate information based on your configuration and SSL settings (see “Using <code>certinfo</code> ” on page 208)
<code>changepw</code>	Changes the Identity Synchronization for Windows configuration password (see “Using <code>changepw</code> ” on page 208)
<code>importcnf</code>	Imports an exported Identity Synchronization for Windows version 1.0 configuration XML document (see “Using <code>importcnf</code> ” on page 210)
<code>prepds</code>	Prepares a Sun Java System Directory Server source for use by Identity Synchronization for Windows (see “Using <code>prepds</code> ” on page 210)
<code>printstat</code>	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 <code>printstat</code> ” on page 214)
<code>resetconn</code>	Resets connector states in the configuration directory to <i>uninstalled</i> (see “Using <code>resetconn</code> ” on page 215)
<code>resync</code>	Links and resynchronizes existing users or groups and pre-populates directories as part of the installation process (see “Using <code>resync</code> ” on page 215)
<code>groupsync</code>	Synchronizes group information between users and groups from one directory source to another (see “Using <code>groupsync</code> ” on page 218)
<code>accountlockout</code>	Synchronizes account lockout and unlockout between Directory Server and Active Directory sources (see “Using <code>accountlockout</code> ” on page 218)
<code>dspluginconfig</code>	Configures and unconfigures Directory Server plugin on a specified host (see “Using <code>dspluginconfig</code> ” on page 219)

TABLE A-4 Quick Reference to `idsync` Subcommands (Continued)

Subcommand	Purpose
<code>startsync</code>	Starts synchronization (see “Using <code>startsync</code>” on page 219)
<code>stopsync</code>	Stops synchronization (see “Using <code>stopsync</code>” on page 220)

Using `certinfo`

You can use the `certinfo` subcommand to display certificate information based on your configuration and SSL settings. This information can help you determine which certificates must be added for each connector and/or Directory Server Plug-in certificate database.

To display certificate information, open a terminal window (or Command Window) and type the **`idsync certinfo`** command as follows:

```
idsync certinfo [bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z]  
[-P cert-db-path] [-m secmod-db-path]
```

Note – Because the `certinfo` subcommand does not have access to the connectors’ and Directory Server’s certificate databases, some of the required steps it lists might have already been performed.

For example:

```
idsync certinfo -w admin-password -q configuration-password
```

Note – For detailed information about the `certinfo` arguments, review [“Common Arguments to the `Idsync` Subcommands” on page 203](#).

Using `changepw`

You can use the `changepw` subcommand to change the Identity Synchronization for Windows configuration password.

▼ To Change the Configuration Password for Identity Synchronization for Windows:

- 1 Stop all Identity Synchronization for Windows processes (for example, System Manager, Central Logger, Connectors, Console, Installers/Uninstallers).

- 2 After stopping all the processes, back up the `ou=Services` tree by exporting the configuration directory to `ldif`.

- 3 Type the `idsync changepw` command as follows:

```
idsync changepw [-D bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password  
[-Z] [-P cert-db-path] [-m secmod-db-path]  
-b new password | - [-y]
```

For example:

```
idsync changepw -w admin password -q old config password -b -q new config password
```

The following arguments are unique to `changepw`:

Argument	Description
-b password	Specifies a new configuration password. The - value reads the password from standard input (STDIN).
[-y]	Does not prompt for command confirmation.

- 4 Respond to the messages that display in the terminal window. For example,

```
Are you sure that want to change the configuration password (y/n)? yes  
Before restarting the system -  
you must edit the $PSWHOME/resources/SystemManagerBootParams.cfg file  
and change the 'deploymentPassword' to the new value.
```

SUCCESS

- 5 You must modify the `SystemManagerBootParams.cfg` file before restarting the system.

The `SystemManagerBootParams.cfg` file in `$PSWHOME\resources` (where `$PSWHOME` is the *isw-installation directory*) contains the configuration password the system manager uses to connect to the configuration directory.

For example, you would change the password value as follows:

```
From: Parameter name="manager.configReg.deploymentPassword" value="oldpassword"/>  
To: Parameter name="manager.configReg.deploymentPassword" value="newpassword"/>
```

- 6 If the program reports any errors, restore the configuration directory using the `ldif` from ["Using changepw" on page 208](#) and then try again. The most likely reason for an error is that the Directory Server hosting the configuration directory became unavailable during the password change.

Using importcnf

After installing Core ([Chapter 3, “Installing Core”](#)), use the `idsync importcnf` subcommand to import your exported Identity Synchronization for Windows version 1.0 or 1.1 (SP1) configuration XML file, which contains Core configuration information.

To import your version 1.0 configuration XML file, open a terminal window (or Command Window) and type the **idsync importcnf** command as follows:

```
idsync importcnf [-D bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z] [-P cert-db-path]  
[-m secmod-db-path] -f filename [-n]
```

For example:

```
idsync importcnf -w admin_password -q configuration_password -f "MyConfig.cfg"
```

The following arguments are unique to `importcnf`:

TABLE A-5 idsync importcnf Arguments

Argument	Description
-f <i>filename</i>	Specifies the name of your configuration XML document.
-n	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Note – For detailed information about other `importcnf` arguments, review [“Common Arguments to the Idsync Subcommands” on page 203](#).

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 210](#) 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 recreate the Retro-Changelog database for the preferred Sun directory source for any reason, the default access control settings will not allow the Directory Server Connector to read the database contents.

To restore the access control settings for the Retro-Changelog database, run `idsync prepds` or click the Prepare Directory Server button after selecting the appropriate Sun directory source in the Console.

You can configure your system to automatically remove (or *trim*) Changelog entries after a specified period of time. From the command line, modify the `nsslapd-changelogmaxage` configuration attribute in `cn=Retro Changelog Plug-in`, `cn=plugins`, `cn=config`:

`nsslapd-changelogmaxage: IntegerTimeunit`

Where:

- **Integer** is a number.
- **Timeunit** is `s` for seconds, `m` for minutes, `h` for hours, `d` for days, or `w` for weeks. (There should be no space between the Integer and Timeunit variables.)

For example, `nsslapd-changelogmaxage: 2d`

For more information, see the “Managing Replication” chapter in the Sun Java System Directory Server 5 2004Q2 Administration Guide.

- You can use *Administrative* credentials to prepare a *secondary* server.

Be sure to plan your Identity Synchronization for Windows configuration *before* running `idsync prepds` because you must know which hosts and suffixes you will be using.

Running `idsync prepds` on a Directory Server suffix where the Directory Server Connector and Plug-in are already installed, configured, and synchronizing will result in a message asking you to install the Directory Server Connector. Disregard this message.

To prepare a Sun Java System Directory Server source, open a terminal window (or a Command Window) and type the **`idsync prepds`** command as follows:

For single host:

```
idsync prepds [-h <hostname>] [-p <port>] [-D <Directory Manager DN>] -w <password>
-s <database suffix> [-x] [-Z] [-P <cert db path>] [-m <secmod db path>]
```

For multiple hosts:

```
idsync prepds -F <filename of Host info> -s <root suffix> [-x] [-Z]
[-P <cert db path>] [-m <secmod db path>] [-3]
```

For example:

```
isw-hostname\bin>idsync prepds -F isw-hostname\samples\Hosts.xml \
-s ou=isw_data
```

Note – The -h, -p, -D, -w, and -s arguments are redefined (as described in the following table) for the prepds subcommand only. In addition, the -q argument does not apply.

“Using prepds” on page 210 describes the arguments that are unique to idsync prepds.

TABLE A-6 prepds Arguments

Argument	Description
-h <i>name</i>	Specifies the DNS name of the Directory Server instance serving as the preferred host.
-p <i>port</i>	Specifies port number for Directory Server instance serving as preferred host. (Default is 389.)
-j <i>name (optional)</i>	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 <i>port (optional)</i>	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 <i>dn</i>	Specifies the distinguished name of the Directory Manager user for the preferred host.
-w <i>password</i>	Specifies a password for the Directory Manager user for the preferred host. The - value reads the password from standard input (STDIN).
-E <i>admin-DN</i>	Specifies the distinguished name of the Directory Manager user for the secondary host.
-u <i>password</i>	Specifies a password for the Directory Manager user for the secondary host. The - value reads the password from standard input (STDIN).

TABLE A-6 prepd Arguments (Continued)

Argument	Description
-s <i>rootsuffix</i>	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 <code>dspswuserlink</code> attribute to the database.
-F <i>filename of Host info</i>	Specifies the filename containing the host information in case of multiple hosts environment.

If you are running `idsync prepd`s in a replicated environment, (for example, where you have a preferred master, a secondary master, and two consumers), you only need to run `idsync prepd`s once for the preferred and secondary masters.

▼ To run `idsync prepd`s

- 1 Ensure that Directory Server replication is up and running (if applicable.)
- 2 Run `idsync prepd`s from the Console or from the command line, for example:

```
idsync prepd -h M1.example.com -p 389 -j M2.example.com -r 389.
```

Running the `idsync prepd`s command on M1 accomplishes the following:

- Enables and extends the RCL to capture more attributes (`dspswuserlink` and so forth) RCL is required on M1 only.
- Extends schema.
- Adds `uid=pswconnector,suffix` user with ACIs.
- Adds indexes to the `dspswuserlink` attribute, which puts Directory Server in read-only mode temporarily until the indexing is done.
You can add indexes later to avoid downtime, but you must add indexes *before* installing the Directory Server Connector.

Adds indexes on M2.

Note –

- Replication ensures that Identity Synchronization for Windows copies schema information and the `uid=pswconnector` from the preferred master to the secondary master and both consumers.
 - You must install the Directory Server Connector once. You must install the Directory Server Plug-in in *all* directories.
 - Indexing is required on the preferred and the secondary masters only. (Replication does not push the indexing configuration from the preferred master to the secondary master.)
-

Using `printstat`

You can use the `printstat` subcommand to:

- Display a list of the remaining steps you have to perform to complete the installation and configuration process.
- Print the status of installed connectors, the system manager, and the Message Queue.

Possible status settings include:

- **Uninstalled.** The connector is not installed.
- **Installed.** The connector is installed, but not ready for synchronization because it has not received its runtime configuration yet.
- **Ready.** The connector is ready for synchronization, but is not synchronizing any objects yet.
- **Syncing.** The connector is synchronizing objects.

To print the status of installed Connectors, the System Manager, and the Message Queue open a terminal window (or a Command Window) and enter the **`idsync printstat`** command as follows:

```
idsync printstat [-D bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z]  
[-P cert-db-path] [-m secmod-db-path]
```

For example:

```
idsync printstat -w admin password -q configuration password
```

Using resetconn

You can use the `resetconn` subcommand to reset connector states in the configuration directory to *uninstalled*. For example, if a hardware failure prevents you from uninstalling a connector, use `resetconn` to change the connector’s status to *uninstalled* so you can reinstall that connector.

Note – The `resetconn` subcommand is intended to be used only in the event of hardware or uninstaller failures.

To reset the state of connectors from the command line, open a terminal window (or a Command Window) and type the **idsync resetconn** command as follows:

```
idsync resetconn [-D bind-DN] -w bind-password\> | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z] [-P cert-db-path]  
[-m secmod-db-path] -e directory-source-name [-n]
```

For example:

```
idsync resetconn -w admin password -q configuration_password -e "dc=example,dc=com"
```

“[Using preps](#)” on page 210 describes the arguments that are unique to `resetconn`:

TABLE A-7 idsync resetconn Arguments

Argument	Description
-e <i>dir-source</i>	Specifies the name of the directory source to reset.
-n	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Note – `idsync printstat` can be used to find directory source names.

For detailed information about the other `resetconn` arguments, review “[Common Arguments to the Idsync Subcommands](#)” on page 203.

Using resync

You can use the `resync` subcommand to bootstrap deployments with existing users. This command uses administrator-specified matching rules to

- Link existing entries

- Populate an empty directory with the contents of a remote directory
- Bulk-synchronize attribute values between two existing user populations
- Bulk-synchronize existing groups and the users associated with the groups (when the group synchronization feature is enabled).

Note – For more detailed information about linking and synchronizing users, see [Chapter 1, “Understanding the Product.”](#)

To resynchronize existing users and to pre-populate directories, open a terminal window (or a Command Window) and type the **idsync resync** command as follows:

```
idsync resync [-D bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z] [-P cert-db-path]  
[-m secmod-db-path] [-n] [-f xml filename for linking] [-k] [-a ldap-filter]  
[-l sul-to-sync] [-o Sun | Windows] [-c] [-x]  
[-u][-i ALL_USERS | NEW_USERS | NEW_LINKED_USERS]
```

For example:

```
idsync resync -w admin password -q configuration_password
```

“[Using resync](#)” on [page 215](#) describes the arguments that are unique to resync:

TABLE A-8 idsync resync Usage

Argument	Meaning
-f <i>filename</i>	Creates links between unlinked user entries using one of the specified XML configuration files provided by Identity Synchronization for Windows (see Appendix B, “Identity Synchronization for Windows LinkUsers XML Document Sample”)
-k	Creates links between unlinked users only (does not create users or modify existing users)
-a <i>ldap-filter</i>	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 <code>idsync resync -o Sun -a “uid=*”</code> all Directory Server users that have a uid attribute will be synchronized to Active Directory.
-l <i>sul-to-sync</i>	Specifies individual Synchronization User Lists (SULs) to resynchronize Note: You can specify multiple SUL IDs to resynchronize multiple SULs or, if you do not specify any SUL IDs, the program will resynchronize all of your SULs.

TABLE A-8 idsync resync Usage (Continued)

Argument	Meaning
-o (Sun Windows)	<p>Specifies the source of the resynchronization operation</p> <ul style="list-style-type: none"> ■ 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	<p>Creates a user entry automatically if the corresponding user is not found at destination</p> <ul style="list-style-type: none"> ■ 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)	<p>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.</p> <ul style="list-style-type: none"> ■ ALL_USERS: Forces on-demand password synchronization for all synchronized users ■ NEW_USERS: Forces on-demand password synchronization for newly created users only ■ NEW_LINKED_USERS: Forces on-demand password synchronization for all newly created and newly linked users
-u	<p>Only updates the object cache. No entries are modified.</p> <p>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.</p>
-x	Deletes all destination user entries that do not match a source entry.
-n	Runs in safe mode so you can preview the effects of an operation with no actual changes.

Note –

- Run `idsync resync` with no arguments to view a usage statement.
- For detailed information about the `resync` arguments, review “[Common Arguments to the Idsync Subcommands](#)” on page 203.
- For more information about resynchronizing existing users, review [Chapter 1, “Understanding the Product.”](#)

After running `resync`, check the `resync.log` file in the central audit log. If errors result, consult [Chapter 7, “Troubleshooting Identity Synchronization for Windows,”](#) in *Sun Directory Server Enterprise Edition 7.0 Troubleshooting Guide*.

Using groupsync

You can use the `groupsync` subcommand to synchronize groups between Active Directory and Directory Server.

To enable or disable the Group Synchronization, type **`idsync groupsync`** command.

For example:

```
idsync groupsync -{e/d} -D <bind DN> -w <bind password> [-h <CD hostname>]
[-p <CD port no>] -s <rootsuffix> [-Z] -q <configuration password> -t <AD group type>
```

TABLE A-9 groupsync arguments

Argument	Meaning
-{e/d}	Select e for enabling , and d for disabling the group synchronization.
-t	Specifies the group type at Active Directory. For example, it can be selected as either of "distribution" or "security"

Using accountlockout

You can use the `accountlockout` subcommand to synchronize account lockout and unlockout between Active Directory and Directory Server.

To enable or disable the account lockout, type **`idsync accountlockout`** command.

For example:

```
idsync accountlockout -{e/d} -D <Directory Manager DN> -w <bind-password>
-h <Configuration Directory-hostname> -p <Configuration Directory-port-no>
-s <rootsuffix> [-Z] [-P <cert db path>] [-m <secmod db path>]
-q <configuration password> -t <max lockout attempts>
```

TABLE A-10 accountlockout arguments

Argument	Meaning
-{e/d}	Select e for enabling , and d for disabling the account lockout synchronization.
-t	Specifies the maximum number of lockout attempts that Active Directory Connector performs.

Using dspluginconfig

You can use the `dspluginconfig` subcommand to configure or unconfigure Directory Server plugin on a specified Directory Server data source.

To configure or unconfigure the Directory Server plugin, type **idsync dspluginconfig** command.

For example:

```
idsync dspluginconfig -{C/U} -D <bind DN> -w <bind password | ->
[-h <CD hostname>] [-p <CD port no>] [-s <configuration suffix>]
[-Z] [-P <cert db path>] [-m <secmod db path> ] [-d <ds plugin hostname>]
[-r <ds plugin port>] [-u <ds plugin user>] [-x <ds plugin user password>]
[-o <database suffix>] [-q <configuration password | ->]
```

TABLE A-11 dspluginconfig arguments

Argument	Meaning
-{C/U}	Select C for configuring and U for unconfiguring the Directory Server plugin.
-d	Host name of the Directory Server data source where the plugin needs to be configured
-r	Port number of the Directory Server data source where the plugin needs to be configured
-u	Administrator of the Directory Server data source where the plugin needs to be configured
-x	Password of the administrator of the Directory Server data source where the plugin needs to be configured
-o	Data suffix of the Directory Server data source.

Using startsync

You can use the `startsync` subcommand to start synchronization from the command line.

To start synchronization, open a terminal window (or a Command Window) and type the **idsync startsync** command as follows:

```
idsync startsync [-D bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z]  
[-P cert-db-path] [-m secmod-db-path]
```

For example:

```
idsync startsync -w admin password -q configuration_password
```

“[Using startsync](#)” on page 219 describes the arguments that are unique to startsync.

TABLE A-12 idsync startsync Arguments

Argument	Description
[-y]	Does not prompt for command confirmation.

Note – For detailed information about the other startsync arguments, review “[Common Arguments to the Idsync Subcommands](#)” on page 203.

Using stopsync

You can use the stopsync subcommand to stop synchronization from the command line.

To stop synchronization, open a terminal window (or a Command Window) and type the **idsync stopsync** command as follows:

```
idsync stopsync [-D bind-DN] -w bind-password | -  
[-h Configuration Directory-hostname] [-p Configuration Directory-port-no]  
[-s rootsuffix] -q configuration_password [-Z]  
[-P cert-db-path] [-m secmod-db-path]
```

For example:

```
idsync stopsync -w admin password -q configuration_password
```

Note – For detailed information about the stopsync arguments, review “[Common Arguments to the Idsync Subcommands](#)” on page 203.

Using the forcepwchg Migration Utility

Users who change their passwords during migration will have different password in Windows NT and the Directory Server. You can use the `forcepwchg` utility to require a password change for users who changed their passwords during the Identity Synchronization for Windows version 1.0 to version 6.0 migration process.

Note – The `forcepwchg` utility ships with Windows packages only.

Before using `forcepwchg` you must verify the following:

- Be sure you do not configure the 7-bit check Plug-in in Directory Server to enforce 7-bit values for the `userpassword` attribute. Do this using the Directory Server console.
- Be sure that the client you are using for authentication translates the value from your locale to UTF-8 correctly. (For example, the `-i` option for the `ldapsearch` shipped with Directory Server).

▼ To Execute the `forcepwchg` Command line Utility

- 1 Open a Command Prompt window and `cd` to the `Windows migration` directory on the host where you are performing the migration. (The Identity Synchronization for Windows 1.0 NT components such as connector, Change Detector DLL, and Password Filter DLL must be installed on the PDC host.)

- 2 From the `migration` directory, type

```
java -jar forcepwchg.jar [-n] [-a] [-t <
time_specification>]
```

For example,

```
forcepwchg.jar -n -a forcepwchg.jar -t 33m
```

“Using the `forcepwchg` Migration Utility” on page 221 describes the arguments that are unique to `forcepwchg`:

Option	Description
-n	Specifies <i>preview mode</i> . In the preview mode, the utility prints out the names of all normal users except: <ul style="list-style-type: none">■ 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 <i>time_specification</i>	Forces all users who changed passwords in the past <i>time_specification</i> to change their passwords. Where <i>time_specification</i> can have the following form: <ul style="list-style-type: none">■ <i>number</i>: Number of seconds (for example, -t 30)■ <i>number m</i>: Number of minutes (for example, -t 25m)■ <i>number h</i>: 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.

Identity Synchronization for Windows LinkUsers XML Document Sample

This appendix provides two sample XML configuration documents that you can use with the `idsync resync` subcommand to link existing users in your deployment.

Both of the following files are available in the `samples1` subdirectory where you installed Core:

- “[Sample 1: linkusers-simple.cfg](#)” on page 223 (an example of a common and simple configuration)
- “[Sample 2: linkusers.cfg](#)” on page 224 (a more-complex configuration example that shows the full power of specifying linking criteria)

You can modify the samples to suit your environment. Both files contain comments that explain how to modify the samples to link your users — including how to link users in multiple SULs.

Sample 1: linkusers-simple.cfg

```
<!--
  Copyright 2004 Sun Microsystems, Inc. All rights reserved
  Use is subject to license terms.
--\>
<!--
  This xml file is used to link Windows and
  Sun Directory Server users from the commandline.
  It is passed to the 'idsync resync'
  script as the -f option. This is a simple file
  that links users in the SUL1 synchronization user list
  that have the same login name, that is the Directory Server
  uid attribute matches the Active Directory
  samaccountname attribute. For more complex
  matching rules, see the linkusers.cfg sample.
--\>
```

```
<UserLinkingOperationList\>
  <UserLinkingOperation parent.attr="UserLinkingOperation"
    sulid="SUL1"\>
    <UserMatchingCriteria parent.attr="UserMatchingCriteria"\>
      <AttributeMap parent.attr="AttributeMap"\>
        <AttributeDescription parent.attr="SunAttribute"
          name="uid"/\>
        <AttributeDescription parent.attr="WindowsAttribute"
          name="samaccountname"/\>
      </AttributeMap\>
    </UserMatchingCriteria\>
  </UserLinkingOperation\>
</UserLinkingOperationList\>
```

Sample 2: linkusers.cfg

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

    Copyright 2004 Sun Microsystems, Inc.
    All rights reserved
    Use is subject to license terms.

--\>
<!--
    This xml file is used to link Windows
    and Sun Directory Server users from
    the command line. It is passed to the
    \qidsync resync\q script as the -f option.

--\>
<!--
    The following parameters allowLinkingOutOfScope:
    if true, then Windows users can be
    linked to Sun Directory Server users
    that are outside of the users\q Synchronization
    User List. Default is false.

--\>
<UserLinkingOperationList allowLinkingOutOfScope="false"\>

<!--
    UserLinkingOperation encapsulates the configuration
    of a single SUL to link. It includes the SUL ID
    and a list of attributes to match.
    A separate UserLinkingOperation must
    be specified for each SUL being linked.

--\>
<UserLinkingOperation parent.attr="UserLinkingOperation" sulid="SUL1"\>
```



```

<!--
    UserMatchingCriteria encapsulates a
    list of attributes that must match for a user to be linked. --\>
<!--

    For two users to match using this UserMatchingCriteria,
    they must have the same givenName and the same sn. --\>
<UserMatchingCriteria parent.attr="UserMatchingCriteria">
    <AttributeMap parent.attr="AttributeMap">
        <AttributeDescription parent.attr="SunAttribute" name="sn"/>
        <AttributeDescription parent.attr="WindowsAttribute" name="sn"/>
    </AttributeMap>    <AttributeMap parent.attr="AttributeMap">
        <AttributeDescription parent.attr="SunAttribute" name="givenName"/>
        <AttributeDescription parent.attr="WindowsAttribute"
name="givenName"/>    </AttributeMap></UserMatchingCriteria>

<!--
    Multiple UserMatchingCriteria can be specified for a single SUL.
    They are treated as a logical OR. In this example,
    the givenName\qs and sn\qs must match (see above)) OR
    (the employee(Number|ID) must match),
    for the user to be linked. Notice that attribute
    that is specified, employeeNumber,
    is the name of the DS attribute. --\>
<!--
    This UserMatchingCriteria is commented out because
    employeeNumber is not an indexed attribute in DS.
    All attributes used in a UserMatchingCriteria
    should be indexed.
<UserMatchingCriteria parent.attr="UserMatchingCriteria">
    <AttributeMap parent.attr="AttributeMap">
        <AttributeDescription parent.attr=
"SunAttribute" name="employeeNumber"/>
        <AttributeDescription parent.attr=
"WindowsAttribute" name="employeeID"/>
    </AttributeMap>
</UserMatchingCriteria>
--\>
</UserLinkingOperation>
<!--
    When multiple SULs are linked, a separate UserLinkingOperation
    is specified for each.
    As shown here, each UserLinkingOperation can use different
    UserMatchingCriteria: in this example, users in SUL2 are
    only linked if their sn and employeeNumber match.
    Note: this UserLinkingOperation is currently
    commented out because the example configuration

```

```
    only has a single SUL.
<UserLinkingOperation parent.attr="UserLinkingOperation" sulid="SUL2">
  <UserMatchingCriteria parent.attr="UserMatchingCriteria">
    <AttributeMap parent.attr="AttributeMap">
      <AttributeDescription parent.attr="SunAttribute" name="sn"/>
      <AttributeDescription parent.attr="WindowsAttribute" name="sn"/>
    </AttributeMap>
    <AttributeMap parent.attr="AttributeMap">
      <AttributeDescription parent.attr=
        "SunAttribute" name="employeeNumber"/>
      <AttributeDescription parent.attr=
        "WindowsAttribute" name="employeeID"/>
    </AttributeMap>
  </UserMatchingCriteria>
</UserLinkingOperation>
--\>
</UserLinkingOperationList>
```

Running Identity Synchronization for Windows Services as Non-Root on Solaris

You must have root privileges to install and to run Identity Synchronization for Windows services on Solaris and Red Hat systems.

However, after installing the product you can configure the software to run the program services as a non-root user.

Running Services as a Non-root User

Note – To run services as non-root, you must change the permissions for all directories under the Identity Synchronization for Windows instance directory. The *default* directory is `/var/opt/SUNWisw`.

▼ To Run services as a Non-root User

Although you must be root to install and to run Identity Synchronization for Windows services, you can configure the software to run the program services as a non-root user.

- 1 **(Optional) Use the UNIX `useradd` command to create a user account for Identity Synchronization for Windows.**

You also can use a nobody user to run services. The remaining examples in this procedure assume you created a user called `iswuser`.

- 2 **To install a Sun Java System Directory Server Connector, you must choose a non-privileged port for the Connector during installation.**

For example, ports larger than 1024 are acceptable. Port 1389 is recommended for LDAP when the server is running as a non-root user. Port 1636 is recommended for LDAP over SSL.

Note – You must execute all commands in the remaining steps as root.

- 3** After installing all components, execute the following command to stop Identity Synchronization for Windows:

```
/etc/init.d/isw stop
```
- 4** You must update the ownership of the instance directory. For example, if you installed the product in `/var/opt/SUNWisw`.

```
chown -R iswuser /var/opt/SUNWisw
```



```
chown -R iswuser /opt/SUNWisw
```
- 5** In a text editor, open the `/etc/init.d/isw` file and replace the following line:

```
"$EXEC_START_WATCHDOG" "$JAVA_PATH" "$INSTALL_DIR" "$CONFIG_DIR"
```


with the following:

```
su iswuser -c "$EXEC_START_WATCHDOG '$JAVA_PATH' '$INSTALL_DIR' '$CONFIG_DIR'"
```
- 6** Execute the following command to restart the service:

```
/etc/init.d/isw start
```
- 7** Execute the following command to verify that the components are running using the assigned user's userid:

```
ps -ef | grep iswuser
```

Defining and Configuring Synchronization User Lists for Identity Synchronization for Windows

This appendix provides supplemental information about Synchronization User List (SUL) definitions and explains how to configure multiple domains. The information is organized as follows:

- [“Understanding Synchronization User List Definitions” on page 229](#)
- [“Configuring Multiple Windows Domains” on page 231](#)

Understanding Synchronization User List Definitions

Every Synchronization User List (SUL) contains two definitions — one to identify which Directory Server users to synchronize and the other to identify which Windows users to synchronize.

Each definition identifies which users in a directory to synchronize, which users to exclude from synchronization, and where to create new users.

Note – The objectclasses you select using the Identity Synchronization for Windows Console also determine which users will be synchronized. The program synchronizes only those users that have the selected objectclass, which includes any users that have a subclass of the selected objectclass.

For example, if you select the `organizationalPerson` objectclass, then Identity Synchronization for Windows will synchronize users with the `inetorgperson` objectclass because it is a subclass of the `organizationalPerson` objectclass.

[“Understanding Synchronization User List Definitions” on page 229](#) describes the components of an SUL definition:

TABLE D-1 SUL Definition Components

Component	Definition	Applicable		
		Sun	AD	NT
Base DN	<p>Defines the parent LDAP node of all users to be synchronized.</p> <p>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, <code>ou=sales,dc=example,dc=com</code>.</p>	Yes	Yes	No
Filter	<p>Defines an LDAP-like filter used to include or exclude users from a Synchronization User List. The filter can include the <code>&</code>, <code> </code>, <code>!</code>, <code>=</code>, and <code>*</code> operators. The <code>\>=</code> and <code><=</code> operators are not supported. All comparisons are done using case-insensitive string comparisons.</p> <p>For example, <code>(& (employeeType=manager) (st=CA))</code> will include managers in California only.</p>	Yes	Yes	Yes
Creation Expression	<p>Defines the parent DN and naming attribute of newly created users (applicable only when you enable creates).</p> <p>The creation expression must include the base DN of the Synchronization User List. For example, <code>cn=%cn%,ou=sales,dc=example,dc=com</code>. (Where the <code>%cn%</code> token is replaced with a value from the user entry being created.)</p>	Yes	Yes	No

Note – To synchronize users in a Sun Java System Directory Server with multiple Active Directory domains, you must define at least one SUL for each Active Directory domain.

When Group Synchronization is enabled, the following are important:

1. The creation expression supported at Active Directory is `cn=%cn%`.
2. The creation expression must contain valid attribute names belonging to the group objectclass since the creation expression is common to both user as well as group.

For example:

The attribute `sn` is not part of the `groupofuniquenames` objectclass at the Directory Server. Hence the following creation expression would be invalid for a group object. (Though it would work fine for user.)

```
cn=%cn%.%sn%
```

3. The attribute used in the creation expression must be provided with a value for every user/group entry created. If the value is not provided then the user/group object will not synchronize and an appropriate message will be logged in the central log.
-

When you define multiple SULs, Identity Synchronization for Windows determines membership in an SUL by iteratively matching each SUL definition. The program examines the SUL definitions with more-specific base DN's first. For example, the program tests a match against `ou=sales,dc=example,dc=com` before testing `dc=example,dc=com`.

If two SUL definitions have the same base DN and different filters, then Identity Synchronization for Windows cannot determine automatically which filter should be tested first, so you must use the Resolve Domain Overlap feature to order the two SUL definitions. If a user matches the base DN of an SUL definition but does not match any filters for that base DN, then the program will exclude that user from synchronization — even if that user matches the filter for a less-specific base DN.

Configuring Multiple Windows Domains

To support synchronizing multiple Windows domains to the same Directory Server container (such as `ou=people,dc=example,dc=com`), Identity Synchronization for Windows uses “synthetic” Windows attributes that contain domain information.

- For Active Directory domains, Identity Synchronization for Windows sets the `activedirectorydomainname` attribute to the Active Directory domain name (such as `east.example.com`) before synchronizing the entry to the Directory Server.
- For Windows NT domains, Identity Synchronization for Windows sets the `user_nt_domain_name` attribute to the Windows NT domain name (such as `NTEXAMPLE`) before synchronizing the entry to the Directory Server.

While these attributes do not actually appear in the Windows user entries, they are available for synchronization in the Identity Synchronization for Windows Console and can be mapped to a Directory Server user attribute. Once Identity Synchronization for Windows maps the domain attributes, they will be set in the Directory Server entries during synchronization and can be used in Synchronization User List (SUL) filters.

The following example illustrates how Identity Synchronization for Windows uses these attributes. This example assumes that three Windows domains (two Active Directory domains and one Windows NT domain) will be synchronized with a single Directory Server instance.

▼ To Configure Multiple Windows Domains

- 1 **Users in the Active Directory** `east.example.com` **domain will be synchronized to the Directory Server in** `ou=people,dc=example,dc=com`.
- 2 **Users in the Active Directory** `west.example.com` **domain will be synchronized to the Directory Server in** `ou=people,dc=example,dc=com`.
- 3 **Users in the Windows NT** `NTEXAMPLE` **domain will be synchronized to the Directory Server in** `ou=people,dc=example,dc=com`.

When you create or modify a Directory Server user, the program uses the SUL filters to determine in which Windows domain to synchronize the user (because each Directory Server SUL has the same base DN, `ou=people,dc=example,dc=com`). The `activedirectorydomainname` and `user_nt_domain_name` attributes make constructing these filters easy.

To construct a filter from the Attributes tab on the Console:

- 4 **Map the Directory Server** `destinationindicator` **attribute to the Active Directory** `activedirectorydomainname` **attribute and to the Windows NT** `user_nt_domain_name` **attribute.**
- 5 **Configure one SUL for each Windows domain as follows:**

`EAST_SUL`

Sun Java System Directory Server definition

Base DN: `ou=people,dc=example,dc=com`

Filter: `destinationindicator=east.example.com`

Creation Expression: `cn=%cn%,ou=people,dc=example,dc=com`

Active Directory definition (`east.example.com`)

Base DN: `cn=users,dc=east,dc=example,dc=com`

Filter: `<none>`

Creation Expression: `cn=%cn%,cn=users,dc=east,dc=example,dc=com`

WEST_SUL

Sun Java System Directory Server definition

Base DN: ou=people,dc=example,dc=com

Filter: destinationindicator=west.example.com

Creation Expression: cn=%cn%,ou=people,dc=example,dc=com

Active Directory definition (west.example.com)

Base DN: cn=users,dc=west,dc=example,dc=com

Filter: <none\>

Creation Expression: cn=%cn%,cn=users,dc=west,dc=example,dc=com

NT_SUL

Sun Java System Directory Server definition

Base DN: ou=people,dc=example,dc=com

Filter: destinationindicator=NTEXAMPLE

Creation Expression: cn=%cn%,
ou=people,dc=example,dc=com

Windows NT definition (NTEXAMPLE)

Base DN: NA

Filter: <none\>

Creation Expression: NA

Notice that each Directory Server SUL definition has the same base DN and creation expression, but the filters indicate the domain of the corresponding Windows user entry.

To further illustrate how these settings allow Directory Server user entries to synchronize with separate Windows domains, consider this test case:

- 6 Create cn=Jane Test,cn=users,dc=example,dc=com in the Active Directory east.example.com domain.**
- 7 Identity Synchronization for Windows creates the user entry cn=Jane Test,ou=people,dc=example,dc=com in the Directory Server with destinationindicator=east.example.com.**
- 8 Modify the cn=Jane Test,ou=people,dc=example,dc=com entry in the Directory Server.**
- 9 Because Jane Test's destinationindicator attribute is east.example.com, her entry will match the EAST_SUL Synchronization User List filter, and the modification will be synchronized to the east.example.com Active Directory domain.**

This example assumes that Identity Synchronization for Windows is synchronizing user creations from Windows to the Directory Server. If this is not the case, you can run the `idsync resync` command to set the `destinationindicator` attribute.

Note – When you use `idsync resync -f` in a deployment with multiple SULs, you probably will have to set the `allowLinkingOutOfScope` option to `true` in the linking configuration file. See [Appendix B, “Identity Synchronization for Windows LinkUsers XML Document Sample”](#)

The example uses an existing attribute in `inetorgperson`, `destinationIndicator`, which might be used for other purposes. If this attribute is already in use or if 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.

Identity Synchronization for Windows

Installation Notes for Replicated Environments

Identity Synchronization for Windows 6.0 supports synchronizing users in a single replicated suffix.

Note – This appendix summarizes procedures used to configure and secure a multimaster replication (MMR) deployment. The information is taken directly from the [Sun Directory Server Enterprise Edition 7.0 Administration Guide](#) — and is not Identity Synchronization for Windows - specific.

Designing and implementing an MMR deployment is *complex*. Refer to the [Sun Directory Server Enterprise Edition 7.0 Deployment Planning Guide](#) to plan your deployment and the [Sun Directory Server Enterprise Edition 7.0 Administration Guide](#) to implement the deployment.

This appendix is organized into the following sections:

- “Configuring Replication” on page 235
- “Configuring Replication Over SSL” on page 237

Configuring Replication

Note – In multimaster replication (MMR) environments, Identity Synchronization for Windows allows you to specify a preferred and secondary master servers for any given Sun directory source.

Directory Server supports n-way MMR (where you can change the replicated database at any of the 'n' masters configured). When you install the plug-in at the preferred master, you must select the *Other* host type and enter Directory Server instance's parameters manually during plug-in installation.

The following steps assume you are replicating a single suffix. If you are replicating more than one suffix, you may configure them in parallel on each server. In other words, you may repeat each step to configure replication on multiple suffixes.

▼ To Configure any Replication Topology

- 1 Define a replication manager entry on all servers except single masters (or use the default replication manager on all servers.)
- 2 On all servers containing a dedicated consumer replica:
 - a. Create an empty suffix for the consumer replica.
 - b. Enable the consumer replica on the suffix through the replication wizard.
 - c. Optionally, configure the advanced replica settings.
- 3 On all servers containing a hub replica, if applicable:
 - a. Create an empty suffix for the hub replica.
 - b. Enable the hub replica on the suffix through the replication wizard.
 - c. Optionally, configure the advanced replica settings.
- 4 On all servers containing a master replica:
 - a. Choose or create a suffix on one of the masters that will be the master replica.
 - b. Enable the master replica on the suffix through the replication wizard.
 - c. Optionally, configure the advanced replica settings.
- 5 Configure the replication agreements on all supplier replicas, in the following order:
 - a. Between masters in a multimaster set.
 - b. Between masters and their dedicated consumers.
 - c. Between masters and hub replicas.

Optionally, you can configure fractional replication at this stage.

- 6 Configure replication agreements between the hub replicas and their consumers.
- 7 For multimaster replication, initialize all masters from the same master replica containing the original copy of the data. Initialize the hub and consumer replicas.

Configuring Replication Over SSL

Note – In this procedure, all references are chapters in the *Sun Directory Server Enterprise Edition 7.0 Administration Guide*.

▼ To Configure Directory Servers Involved in Replication so that all Replication Operations Occur Over an SSL Connection

- 1 **Configure both the supplier and consumer servers to use SSL.**

Refer to Chapter 11, “Managing Authentication and Encryption” for details.

Note –

- Replication over SSL will fail if the supplier server certificate is an SSL server-only certificate that cannot act as a client during an SSL handshake.
 - Replication over SSL is currently unsupported with self-signed certificates.
-

- 2 If replication is not configured for the suffix on the consumer server, enable it as described in Chapter 8, “Enabling a Consumer Replica.”
- 3 Follow the procedure in Chapter 8, “Advanced Consumer Configuration,” to define the DN of the certificate entry on the consumer as another replication manager.
- 4 If replication is not configured for the suffix on the supplier server, enable it as described in Chapter 8, “Enabling a Hub Replica” or “Enabling a Master Replica.”
- 5 On the supplier server, create a new replication agreement to send updates to the consumer on the secure SSL port. Follow the procedure in Chapter 8, “Creating Replication Agreements,” for

detailed instructions. Specify a secure port on the consumer server and select the SSL option of either using a password or a certificate. Enter a DN for the SSL option that you chose, either a replication manager or a certificate.

After you finish configuring the replication agreement, the supplier will send all replication update messages to the consumer over SSL and will use certificates if you chose that option. Customer initialization will also use a secure connection if performed through the console using an agreement configure for SSL.

Configuring Identity Synchronization for Windows in an MMR Environment

▼ To Configure Identity Synchronization for Windows in an MMR Environment

- 1 From the Identity Synchronization for Windows Console, specify a preferred master and secondary master servers for the suffix to be synchronized. (Review [“Creating a Sun Java System Directory Source” on page 85](#))
You do not have to provide information about other Directory Servers in your topology.
- 2 Prepare the preferred master and secondary master servers from the Console or using the `idsync prepds` command line utility. (Review [“Preparing Sun Directory Source” on page 91](#))
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 140](#))
- 4 Configure the Directory Server Plug-in on the preferred master, the secondary masters, and every other Directory Server instance that manages users in the replicated suffix (Review [“Using dspluginconfig” on page 219](#))

Index

Numbers and Symbols

3DES keys, 171

A

access rights, 98, 173, 177, 211

account lockout, 125

accountlockout, Arguments, Description, Syntax, 218

accounts

- built-in, 222

- creating, 55, 141, 227

ACIs, 177, 210

activations, 116

Active Directory

- advanced security options, 100, 171

- attributes, 50, 105, 115

- certificate database, 101

 - importing certificates, 183

- certificates, 99, 101, 171, 178, 183

- change detection, 35

- component distribution example, 43

- configuring Core, 61

- configuring SSL, 59, 91

- connector distribution, 137

- connector-domain controller communication, 40

- connectors, installing, 146-148

- creating directory sources, 95

- creating SULs, 129

- creation expressions, 132

- deployments, 95

- detecting changes, 35

Active Directory (*Continued*)

- directories, 50

- directory sources, 95, 141

- domain controllers, 40, 42, 99, 100, 102

- domains, 95, 97, 231

- editing attributes, 115

- editing domain controller configuration

 - parameters, 102

- enabling secure communication, 91

- failover servers, 100

- global catalogs, 61, 95, 96

- hosts, 96, 97

- installing connectors, 32, 146-148

- linking users, 153, 154

- mapping attributes, 105

- multiple domains, 231

- object creation flow, 111

- object deletions flow, 128

- objectclasses, 50

- on-demand password synchronization, 37, 40, 153

- password policies, 53, 55

- physical deployment, 42

- pre-existing users, 157

- Primary Domain Controller FSMO role owner, 99

- propagating passwords, 59

- resync interval, 102

- sample deployment example, 40

- security options, 100

- selecting attributes, 105

- sources

 - creating, 84

- special users, 157

Active Directory (*Continued*)

- SSL, using, 96, 101, 170, 171, 183
- supported versions, 23
- synchronization settings, 41, 50
- synchronizing activations/inactivations, 117
- synchronizing attributes, 91, 105
- synchronizing deletions, 128
- synchronizing passwords, 40, 53, 91
- synchronizing users, 154
- trusted certificates, 101, 171, 178
- user authentication failure, 39
- user DNs, 96
- using multiple domain controllers, 99
- using SSL, 96, 101, 170, 171, 183

adding

- attribute values, 115
- certificates, 188, 208
- configuration data to Directory Server, 76
- credentials to Administrators group, 176
- directory sources, 84, 94, 104
- indexes, 213
- users to Active Directory, 55

Administration Server

- enabling SSL communication, 72
- installing, 70
- installing Core, 32
- URL location, 78

administrators

- credentials/privileges, 60, 62, 72, 175
- filtering from SULs, 131
- linking users, 153
- preparing Directory Server, 92, 211
- providing (bind) distinguished name, 87, 96
- restricting access, 177
- resynchronizing directory sources, 152
- running uninstall.cmd scripts, 162
- user distinguished names, 96

advanced security options, specifying, 100**alias directory, 183****aliases, certificate, 177****arguments**

- certinfo, 180
- changepw subcommand, 209
- command line utilities, 203

arguments (*Continued*)

- forcepwchg, 221
- importcnf, 205
- password, 205
- preps, 212
- printstat, 214
- resetconn, 215
- resync, 154, 156, 216, 218
- stopsync, 220

attribute modification flow, 116**attributes**

- AvoidPdcOnWan, 99
- creating parameterized default values, 52
- creation, 51
- description, 51
- dspswuserlink, 153, 213
- editing, 115
- indexing, 157
- inetorgperson, 52
- mandatory creation, 51, 107
- mapping, 52, 105
- naming, 129
- nsAccountLock, 118-119
- objectguid, 153
- PwdLastSet, 37
- resynchronizing, 153
- selecting, 50, 105, 110
- significant, 51
- synchronizing user entry, 61, 105
- types, 51
- uid, 155
- user, 52
- USNchanged, 35, 37

audit.log, 59

- description, 29, 191
- linking and resynchronizing results, 218
- location, 190, 198
- purpose, 191

auditing, enabling on Windows NT, 199**authentication**

- client, 221
- connecting to configuration directory, 205
- failures, 39
- on-demand password synchronization, 38

auxiliary objectclasses
 configuring, 51
 removing, 110
 selecting, 109, 110
 AvoidPdcOnWan attribute, 99

B

base DN
 description, 129
 specifying user set domain, 131
 specifying user set domain base DN, 131
 using for multiple SULs, 131
 base64 encoding, 184, 204
 bidirectional synchronization, 24, 29
 broker
 accessing, 177
 Message Queue, 31
 specifying ports, 76
 starting, 159
 stopping, 159
 built-in accounts, 222

C

CA certificates
 adding, 171, 188
 automatic installations, 99
 component requirements, 178
 enabling SSL, 183
 importing, 181
 retrieving, 183, 186, 187
 catalogs, global
 multiple, 95
 protecting, 172
 purpose, 61
 specifying, 95, 97
 central log directories, 16, 190
 central logger
 clogger 100 directories, 191
 description, 28
 local logs, 191
 messages, 190

central logger (*Continued*)
 troubleshooting problems, 191
 centralized
 logs, 190
 system auditing, 25
 certificate database
 adding certificates, 188
 default path, 16
 directories, 186, 188
 required certificates, 180
 retrieving certificates, 183
 specifying location, 205
 certificates
 accepting, 177
 Active Directory, 99, 183
 adding, 188
 aliases, 177
 CA, 171, 178
 certinfo subcommand, 208
 Directory Server, 183
 exporting, 183
 getting information, 207
 importing, 186
 installing, 177
 requiring, 100, 171, 180
 retrieving, 183, 184
 self-signed, 177, 181
 SSL, 100, 171, 177
 using certinfo subcommand, 62, 207
 using certutil, 184
 using idsync certinfo, 180
 validating, 176, 177
 viewing information, 208
 certinfo subcommand
 adding certificates, 208
 arguments, 180
 description, 62, 207
 displaying certificate information, 207
 examples, 208
 syntax, 208
 using, 180
 certutil
 default location, 182
 retrieving certificates, 184

- change detection, 29, 30, 34, 39, 88
- Change Detector subcomponents, 30, 33, 48, 221
- changepw subcommand
 - arguments, 208, 209
 - changing passwords, 208
 - description, 62, 207, 208
 - examples, 208
 - syntax, 208, 209
- changing
 - configuration passwords, 207
 - default schema sources, 108
- channel communication, encrypting, 91
- checklists, 76
 - installation, 63, 65
- clear-text passwords
 - capturing, 35
 - obtaining, 37
 - propagating, 37
 - using Password Filter DLL, 37
- client, authentication, 221
- command line utilities
 - common arguments, 203
 - common features, 203
 - description, 28, 62, 203
 - entering passwords, 205
 - idsync resync, 152
 - using, 62, 203
- commands
 - descriptions, 62
 - imq start, 159
 - isw stop, 159
- communication
 - enabling SSL, 89, 91
 - Last Communication, 197
- components
 - configuration directory, 27
 - Console, 27
 - Core, 27, 47
 - descriptions, 25
 - distribution, 31, 42
 - distribution example, 43
 - IDs, 191
 - local logs, 191
 - logging levels, 193
 - components (*Continued*)
 - messages, 190
 - physical deployment example, 42
- configuration directory
 - administrator name/password, 72, 139
 - connecting to, 205
 - credentials, 175
 - default port, 71
 - description, 27
 - description/explanation, 76
 - encrypting configuration information, 73
 - hostname/port number, 156, 217
 - purpose, 60, 62
 - querying, 86
 - reading/writing to, 27
 - restricting access, 177
 - specifying credentials, 72
 - specifying host/port, 71
 - URL, 60, 71, 139
 - validating certificates, 177
- configuration passwords
 - changing, 207, 208
 - finding, 209
 - protecting, 175
 - specifying, 170
 - using idsync changepw, 208
- Configuration tab, 82
 - description, 83
- configurations, deployment decisions, 60
- configuring
 - activations/inactivations, 117
 - attribute synchronization, 110
 - Core, 22, 60, 63, 79
 - filters, 232
 - log files, 195
 - Message Queue, 75
 - multiple domains, 229
 - multiple suffixes, 236
 - security, 169
 - SSL, 59
 - suffixes, 88
 - To Do list, 47
 - validation, 134

connectors

- Active Directory, 137
- bidirectional synchronization, 29
- description, 29
- detecting changes, 35, 36
- Directory Server, 140
- distribution, 137
- installing, 32, 33, 134, 137
- launching/monitoring, 26
- printing status, 207, 214
- restarting, 29
- states, 207, 215
- troubleshooting, 191
- using idsync printstat, 207
- Watchdog process, 26
- Windows NT, 148

consoles

- configuring Core, 79
- description, 27, 47, 82
- Directory Server, 117
- Identity Synchronization for Windows, 27, 82, 196, 197, 198
- installing, 74
- logging in, 78
- passwords, 73
- reading/writing to configuration directory, 27
- starting, 77
- starting/stopping synchronization, 158
- Sun Java System Console, 80
- uninstalling, 166-167
- viewing logs, 189

Core

- checklists, 63
- components, 25, 47
- configuring, 22, 60, 63, 79
- description, 27
- enabling SSL, 139
- installation privileges, 70
- installing, 32, 60, 63, 70-78
- uninstalling, 161, 164
- Watchdog, 26

creating

- accounts, 55, 141, 227
- Active Directory Sources, 84

creating (*Continued*)

- Active Directory sources, 95
- NT Registry Directory Sources, 84
- parameterized default attribute values, 52
- Retro-Changelog database, 92
- SULs, 52, 54, 129
- Sun Java System Directory Sources, 84, 85-91
- Sun Java System directory sources, 85
- Windows 2003 Server directory sources, 54
- Windows 2003 Server global catalogs, 54
- Windows NT directory sources, 102

creating indexes, 93

creation attributes

- creating, 111
- deleting, 111
- description, 51
- editing, 111
- mandatory, 105, 107
- mapping, 114
- parameterized default values, 52

creation expressions, 53, 132

creation flows

- enabling, 40
- planning configuration, 61
- specifying, 111, 115

credentials/privileges, 72

- configuration directory, 175
- configuration Directory Server, 62
- creating credentials, 175
- Directory Server, 174
- installing Core, 70
- required for connectors, 174
- required for idsync prepd, 211
- specifying, 97
- specifying for configuration directory, 72

custom methods, 119

D

daemons

- starting/stopping, 159
- writing logs, 195

databases

- certificate, 91, 171, 183, 208

databases (*Continued*)

- creating indexes, 93
- object cache, 35
- Retro-Changelog, 92, 94

default locations, 15-17

defaults

- audit/error message lines to show, 198
- base64-encoded values, 204
- broker port, 76
- certutil location, 182
- command line utility arguments, 156
- configuration directory port, 71
- creating parameterized values, 52, 107
- encrypted with 3DES keys, 172
- installation directory for Solaris, 162
- instance directory, 227-228
- keeping logs, 192
- LDAP port, 87
- log directory, 194
- password policies, 53
- Require trusted SSL certificate setting, 101
- resync interval, 94
- resynchronization source, 155
- root suffixes, 88, 204
- SUL name, 130
- synchronization flow, 110
- syslog messages, 195
- writing logs, 195

defining

- multiple domains, 229
- SULs, 229
- users, 52

deleting

- attribute values, 115
- objects, 128

deletions

- specifying flow, 128
- synchronizing, 128

deployments

- Active Directory, 95
- bootstrapping, 49
- component distribution, 31
- examples, 42
- installation/configuration decisions, 60

deployments (*Continued*)

- MMR, 235
- running idsync resync, 49
- single-host, 45
- synchronization requirements, 40
- two-machine scenario, 40

detecting

- activations/inactivations, 117
- changes, 29, 30, 34, 39, 88
- errors, 28

directories

- Active Directory, 50
- alias, 183
- central log, 190
- certificate database, 186, 188
- clogger 100 (central logger), 191
- configuration, 27, 60, 61, 62, 76
- containing centralized logs, 190
- default instance, 227-228
- description/explanation, 50
- installation, 75, 138
- instance, 227-228
- isw-hostname, 162
- logs, 194
- pre-populating, 216
- querying, 86
- resynchronizing sources, 152
- samples1, 223
- specifying installation, 74
- TEMP, 144

Directory Server

- access rights, 98
- accessing via SSL, 204
- attribute modification flow, 116
- change detection, 35
- connectors, installing, 140
- console, 117
- credentials/privileges, 174
- installing connectors, 32, 140
- installing the plugin, 32
- interoperating with Directory Server tools, 117
- objectclasses, 50
- password policies, 54
- preparing, 48, 92, 207, 211

Directory Server (Continued)

- preparing directory sources, 48, 210
- preparing Identity Synchronization for Windows source, 91-95
- propagating passwords, 59, 61
- setup program, 138
- specifying, 89
- synchronizing attributes, 105
- synchronizing passwords, 40
- using custom methods, 119
- using idsync preps, 207

Directory Server Plug-in

- adding certificates, 208
- bidirectional synchronization, 30
- description, 30, 91
- detecting changes, 35
- enabling secure communication, 91, 188
- encrypting passwords, 171
- installing, 32, 91, 137
- logs, 192
- removing, 161
- uninstalling, 161
- using SSL, 91, 188

Directory Server Plugin, installing, 32**directory sources**

- Active Directory, 141
- adding, 84, 94, 104
- creating, 54
- example entries, 140
- linking users, 153
- states, 196

distinguished names

- administrator, 98
- specifying, 96, 98

distributing system components, 31**DLLs**

- NT Change Detector, 192
- Password Filter, 37
- Windows NT, 33

DNs, 96**DNS, domain entries, 88****domain controllers**

- Active Directory, 99, 100
- editing, 102, 104

domain controllers (Continued)

- editing parameters, 102
- failover, 100
- specifying, 98
- using multiple, 99

domains

- Active Directory, 95, 97, 231
- configuring multiple, 229
- multiple, 231
- resolving overlap, 133
- specifying for NT, 103
- user set, 131

downloading, installation program, 68**dspswuserlink attribute, 153, 213****E****editing**

- domain controller configuration parameters, 102
- domain controllers, 102, 104
- mapped attributes, 115

enabling

- SSL communication, 72, 89, 91, 139, 181

encrypting

- 3DES keys, 171
- channel communication, 91
- clear-text passwords, 35
- configuration information, 72, 73
- Message Queue messages, 171, 173
- network traffic, 171

enforcing password policies, 54**equality**

- filters, 131
- indexes, 92, 213

error detection, 28**error.log**

- description, 191
- location, 190, 198
- mapping connector IDs to directory source, 186, 188

errors, validation, 134**examples**

- audit log path, 198
- central log, 191

examples (Continued)

- directory source entries, 140
- forcepwchg command, 221
- idsync certinfo command, 208
- idsync changepw command, 209
- idsync importcnf command, 210
- idsync prepds command, 212
- idsync printstat command, 214
- idsync resetconn command, 215
- idsync resync command, 216
- idsync startsync command, 220
- idsync stopsync command, 220
- log messages, 193
- prepds subcommand, 211
- resync arguments, 156
- user set domain base DN, 131

executable, setup.exe, 138

exporting, Directory Server certificates, 183

F

failover controllers, specifying, 100

failures

- hardware, 207
- uninstallater, 207

features, 24

filtering

- synchronization user lists, 133
- user lists, 131, 230

filters

- configuring, 232
- description, 129
- equality, 131
- LDAP, 53, 65, 205, 216
- presence, 131
- search, 184
- substring, 131
- SUL, 53, 61, 129
- syntax, 131, 230

flow

- defaults, 110
- specifying deletions, 128
- specifying modification, 116

forcepwchg.jar, 221

forcepwchg utility

- arguments, 221
- description, 221
- forcing password changes, 221

FSMO, 99

G

global catalogs, 61

- Active Directory, 95
- creating, 54
- multiple, 95
- protecting, 172
- specifying, 95, 96, 97

global synchronization settings, 41

Group Synchronization, 124, 218

H

hardening security, 175

hardware failures, 207

hashed passwords, 35

help, usage information, 206

high availability description, 39

hostnames

- configuration directory, 156, 217
- server group, 81

hosts

- Active Directory, 96, 97
- specifying, 96

I**Identity Synchronization for Windows**

- Console, 196, 197, 198
- installation, 45
- preparing Directory Server directory sources, 48, 210
- preparing Directory Server source, 91-95
- reliability, 39
- removing, 22, 161
- setup program, 22, 67

Identity Synchronization for Windows (*Continued*)

- uninstalling, 161
- idsync certinfo, 180
 - adding certificates, 208
 - arguments, 208
 - description, 208
 - examples, 208
 - syntax, 208
- idsync changepw
 - arguments, 208
 - changing passwords, 208
 - description, 208
 - examples, 208
 - syntax, 208
- idsync groupsync, Arguments, Description, Syntax, 218
- idsync importcnf
 - arguments, 205, 210
 - description, 207, 210
 - importing configuration files, 210
 - syntax, 210
- idsync prepsd
 - credentials, 211
 - description, 62, 207
 - preparing Directory Server, 48, 207
 - syntax, 212
- idsync printstat
 - arguments, 214
 - description, 214
 - listing install/configuration steps, 214
 - printing status, 214
 - syntax, 214
- idsync resetconn
 - arguments, 215
 - description, 215
 - syntax, 215
- idsync resync, 49
 - argument examples, 156
 - arguments, 215
 - caveats for using, 157
 - description, 215
 - example usages, 157
 - indexes, 157
 - logging results, 157

idsync resync (*Continued*)

- resynchronizing two directory sources, 152
- sample linkusers XML configuration documents, 223
- scripts, 153
- synchronizing existing users, 216
- syntax, 215
- using, 152
- idsync script, executing, 206
- idsync startsync
 - arguments, 219
 - description, 219
 - syntax, 219
- idsync stopsync
 - arguments, 220
 - description, 220
 - syntax, 220
- importcnf subcommand
 - arguments, 205, 210
 - description, 207, 210
- importing
 - CA certificates, 181
 - configuration information, 210
- imq start commands, 159
- imq stop commands, 159
- inactivations, 116
- indexes
 - adding, 213
 - creating, 93
 - creating equality, 92
- indexing attributes, 157
- inetorgperson attribute, 52
- information panel, 47, 76, 83, 144, 147, 197
- install-path*, 16
- installation
 - checklists, 63, 65
 - decisions, 60
 - directories, 138
 - directories, default, 162
 - downloading program, 68
 - specifying directories, 74, 75
 - To Do list, 47, 76
 - viewing logs, 144, 147, 149

installing

- Active Directory connectors, 32, 146-148
 - certificates, 177
 - connectors, 134, 137
 - Core, 32, 60, 70-78
 - Directory Server connectors, 32
 - Directory Server Plug-in, 137
 - Directory Server Plugin, 32
 - Identity Synchronization for Windows, 74
 - subcomponents, 134
 - Windows NT connectors and subcomponents, 33
- instance directory, default, 227-228
- instance-path*, 16
- interoperating, with Directory Server Tools, 117
- isw-hostname* directory, 16
- isw-hostname* directory, 162
- isw start* command, 159
- isw stop* commands, 159

J

- jar files, forcepwhchg, 221
- Java Development Kits, downloading, 67
- Java Home, specifying, 73
- Java Naming and Directory Interface, 15
- java processes
- central logger, 28
 - command line utilities, 28
 - configuration directory, 27
 - connectors, 29
 - Console, 27
 - restarting, 26
 - system manager, 28
 - Watchdog, 26

JRE

- downloading, 67
- verifying Java Home directory, 73

K

- keytool utility, 177

L

- launching connectors, 26
- LDAP
- default port, 87
 - DIT, 61
 - filters, 53, 65, 205, 216
 - ldapsearch, 221
 - query syntax, 131
- ldapsearch, using, 221
- lightweight processes, 30
- linking users
- using idsync resync, 207
 - using XML configuration documents, 216
- linkusers.cfg, 223, 224-226
- linkusers-simple.cfg, 223-224
- LinkUsers XML Document, 223
- local log directory, 16
- local logs, 191
- central logger, 191
 - component, 191
- locating PDC computer names, 103
- logging
- audit/error files, 189
 - central logs, 190
 - checking resync.log, 157
 - day-to-day operations, 189
 - errors, 189
 - log types, 190
 - properly linked users, 157
 - specifying default log directories/files, 194
 - specifying logging levels, 193
 - viewing logs, 144, 147, 149
- logging in, 78
- logs
- audit, 29, 191
 - audit.log, 191
 - Directory Server Plug-in, 192
 - error, 29, 191, 198
 - format, 193
 - local, 191
 - local component logs, 191-192
 - local subcomponent logs, 192
 - location, 198
 - locations, 190

logs (*Continued*)
 reading, 193
 resync, 191
 resync.log, 157
 viewing, 144, 147, 149, 189
 logs directory, 190, 194

M

mandatory creation attributes, 51, 105, 107
 mapping
 attributes, 52, 105, 115
 connector IDs to directory source, 186
 creation attributes, 114
 Message Queue, 15
 accepting certificates, 177
 access controls, 173
 broker, 31
 configuring, 75
 default broker port, 76
 description, 31
 self-signed certificates, 177
 specifying localhost name, 75
 specifying port numbers, 75
 validating certificates, 176
 validating client certificates, 176
 messages
 audit.log, 191
 debug.log, 191
 error.log, 191
 for components, 190
 provided in central logger, 190
 resync.log, 191
 synchronization event, 191
 Microsoft
 certificate server, 99
 Knowledge Base Articles, 99
 migration, using forcepwhg, 221
 MMR
 configuration components, 178
 configuring, 235
 reliable synchronization, 40
 modifications, specifying flow, 116
 monitoring connectors, 26

Multimaster Replication. *See* MMR, 235
 multiple domain controllers, 99
 multiple domains, 229

N

naming attributes, description, 129
 nsAccountLock attribute, 118-119
 NT Change Detector DLLs, 192
 NT Registry Directory Source, 84
 NT SAM
 domain users, 153
 identifiers for linking, 153
 registries, 30
 synchronizing, 33

O

object cache, databases, 35
 objectclasses
 Active Directory, 50
 attributes, 50, 110
 auxiliary, 50
 configuring, 51
 Directory Server, 50
 selecting, 109
 structural, 50
 User, 61
 objectguid attribute, 153
 objects
 configuring activations/inactivations, 117
 deleting, 128
 specifying deletion flow, 128
 specifying modification flow, 116
 on-demand password synchronization, 35, 37, 38, 40, 153
 authentication mechanisms, 38

P

Password Filter subcomponents, 30, 33, 37, 48, 221

- password policies
 - Active Directory, 55
 - default Windows, 53
 - Directory Server, 54
 - enforcing, 54
 - for configuration passwords, 175
 - password synchronization, on demand, 153
 - password synchronization, on-demand, 35, 38
 - passwords
 - arguments, 205
 - changing configuration, 208
 - configuration, 170
 - creating, 111, 115
 - creating accounts without, 55
 - encrypting, 35
 - entering for command line utilities, 205
 - finding, 209
 - hashed, 35
 - on-demand password synchronization, 37, 40, 153
 - propagating changes, 37, 59
 - protecting, 175
 - requiring changes, 221
 - synchronizing, 53
 - PDC
 - FSMO role owner, 99
 - installing connectors and subcomponents, 33
 - locating computer names, 103
 - persistent storage protection, 174
 - port numbers
 - configuration directory, 156, 217
 - defaults, 76
 - specifying Message Queue, 75, 76
 - pre-populating directories, 216
 - prefixes, 88
 - preparing
 - Directory Server, 48, 92, 210
 - preps subcommand
 - arguments, 212
 - credentials, 211
 - description, 62, 207
 - examples, 211
 - preparing Directory Server, 48
 - preparing Directory Servers, 207
 - syntax, 212
 - presence
 - filters, 131
 - indexes, 213
 - Primary Domain Controller., *See* PDC
 - printing connector status, 214
 - printstat subcommand
 - arguments, 214
 - description, 214
 - displaying installation/configuration steps, 207
 - printing connector status, 62, 207
 - syntax, 214
 - privileges/credentials, 60, 72
 - configuration directory, 175
 - configuration Directory Server, 62
 - creating credentials, 175
 - installing Core, 70
 - required for connectors, 174
 - required for idsync preps, 211
 - processes
 - central logger, 28
 - command line utilities, 28
 - configuration directory, 27
 - connectors, 29
 - Console, 27
 - lightweight, 30
 - system manager, 28
 - Watchdog, 26
 - programs, setup, 138
 - propagating
 - new passwords, 111
 - password changes, 37, 59, 116
 - user deletions, 128
 - protecting
 - global catalogs, 172
 - passwords, 175
 - sensitive information, 172
 - protecting sensitive information, 174
 - PwdLastSet attribute, 37
- Q**
- querying
 - configuration directory, 86, 87

R

- reading logs, 193
- Red Hat, running the installation program, 69-70
- reliability, 39
- removing
 - attribute values, 115
 - auxiliary objectclasses, 110
 - Core, 164
 - Directory Server Plug-in, 161
- replication
 - configuring, 178, 236
 - single suffix, 235
 - synchronizing users, 235
- requirements, synchronization, 40
- requiring password changes, 221
- resetconn subcommand, 215
 - arguments, 215
 - description, 215
 - resetting connector states, 62, 207
 - syntax, 215
- resetting
 - connector states, 207, 215
- resolving domain overlap, 133
- resources, finding, 80
- restarting
 - connectors, 29
 - java processes, 26
 - services, 228
 - synchronization, 158
- restricting access, 177
- resync interval
 - default, 94
 - setting for Active Directory connectors, 102
 - setting for Directory Server connectors, 94
 - setting for NT, 104
- resync.log
 - description, 191
 - linking and resynchronizing results, 157, 218
 - location, 190
- resync subcommand, 154, 156, 216, 218, 223
 - arguments, 215
 - bootstrapping deployments, 49
 - description, 215
 - linking and synchronizing users, 152

- resync subcommand (*Continued*)
 - linking/synchronizing users, 62, 207
 - synchronizing existing users, 216
 - syntax, 215
- resynchronizing
 - attributes, 153
 - directory sources, 152
 - users, 207, 215
- retrieving certificates, using certutil, 184
- Retro-Changelog database
 - change detection, 35
 - creating, 92
 - recreating, 94
- role owners, Primary Domain Controller FSMO, 99
- root suffixes
 - default, 88
 - description, 60
 - directory source labels, 48
 - specifying, 72
- running
 - idsync resync scripts, 153
 - out of disk space, 195

S

- safe mode, 154
- samples
 - linkusers.cfg, 224-226
 - linkusers-simple.cfg, 223-224
 - XML configuration documents, 223
- samples1 directory, 223
- SASL Digest-MD5, 38
- schema
 - changing default sources, 108
 - controller, 61
- scripts
 - idsync, 206
 - idsync resync, 153
- secure communication, 91
- Secure Sockets Layer (SSL), 169
- security
 - Active Directory, 100
 - configuring, 169
 - hardening, 175

security (Continued)

- replicated configurations, 178

- self-signed certificates, 177, 181

- serverroot directory, 16

servers

- Administration, 32, 70, 72, 78

- failovers, 100

- finding, 80

- hostnames, 81

services

- restarting, 228

- starting/stopping, 83, 158, 159

- synchronization, 158

- setup.exe, 138

setup programs

- Directory Server, 138

- Identity Synchronization for Windows, 22, 67

- locating, 138

significant attributes

- creating parameterized default values, 52

- description, 51

- single-host deployments, 45

- SLAMD Distributed Load Generation Engine, 14

Solaris

- removing Identity Synchronization for

- Windows, 167

- running the installation program, 68

- starting/stopping daemons, 159

- sources, creating Active Directory, 95

specifying

- Active Directory domains, 97

- attributes, 50, 110

- configuration directory credentials, 72

- configuration directory host/port, 71

- configuration passwords, 170

- creation flows, 111, 115

- credentials, 97

- Directory Server, 89

- domain controllers, 98

- failover controllers, 100

- failover servers, 100

- global catalogs, 95, 96, 97

- hosts, 96

- installation directories, 74

specifying (Continued)

- Java Home, 73

- object deletion flow, 128

- object modification flow, 116

- port numbers, 76

- resync interval, 102

- root suffixes, 72

- user DN, 87, 96

- user DNs, 96

- user set domain base DN, 131

- Windows NT domain names, 103

SSL

- accessing Directory Server, 204

- certificates, 100, 171, 177

- configuring Active Directory, 59, 96, 101

- configuring for Windows, 59

- enabling, 181

- enabling communication, 89, 91, 181

- enabling for Core, 139

- requiring trusted certificates, 100

- selecting ports, 139

- using, 91, 170, 188

- using on Active Directory, 170, 171

starting

- consoles, 77

- daemons, 159

- Message Queue broker, 159

- services, 83, 159

- synchronization, 63, 158, 219

startsync subcommand

- arguments, 219

- description, 219

- starting synchronization, 63, 208

- syntax, 219

- states, directory source, 196

status

- Configuration Validity Status, 133

- connector, 214

- printing connector status, 214

- viewing, 189

- Status tab, 82

- STDIN, reading passwords, 205

stopping

- daemons, 159

stopping (*Continued*)

- Message Queue broker, 159
- services, 83, 159
- synchronization, 63, 158, 220

stopsync subcommand

- arguments, 220
- stopping synchronization, 208
- syntax, 220

storing

- configuration information, 61, 139
- SULs, 133

structural objectclasses

- configuring, 51
- defaults, 51

subcommands

- certinfo, 180, 208
- descriptions, 207
- idsync, 203
- importcnf, 205, 207, 210
- printstat, 214
- resetconn, 215
- resync, 215, 218, 223
- startsync, 219
- stopsync, 220
- using changepw, 208
- using importcnf, 210

subcomponents

- description, 30
- installing, 134

substring filters, 131

suffix/database, 48, 50

suffixes

- configuring, 88
- replicating, 235

SULs

- creating, 52, 54, 129
- defining, 229
- definition components, 129, 229
- definitions, 53
- description, 52, 129
- filtering administrators, 131
- storing, 133

Sun Java System

- Console, 80

Sun Java System (*Continued*)

- creating directory sources, 84, 85-91

Sun Java™, 21

synchronization

- bidirectional, 29
- configuring, 110
- defaults, 110
- event messages, 191
- filtering user lists, 133
- multiple domains, 133
- requirements, 40
- restarting, 158
- settings, 41, 50
- starting, 219
- starting/stopping, 158, 208
- stopping, 220
- using idsync startsync, 208
- using idsync stopsync, 208
- when components become unavailable, 39

Synchronization User Lists. See *SULs*, 129

synchronizing

- activations/inactivations, 117
- attributes, 91, 105
- deletions, 128
- existing users, 49
- NT SAM, 33
- passwords, 40, 53, 91
- user creations, 41
- user entry attributes, 61, 105
- using idsync resync, 207
- with Active Directory, 53

syntax

- changepw subcommand, 209
- forcepwchg command, 221
- idsync, 207
- idsync certinfo command, 208
- idsync changepw command, 209
- idsync importcnf, 210
- idsync prepds command, 212
- idsync printstat command, 214
- idsync resetconn command, 215
- idsync resync command, 216
- idsync startsync command, 220
- idsync stopsync command, 220

- syntax (*Continued*)
 - LDAP filter, 53
 - LDAP query, 131
- system
 - auditing, 25
 - password creation flow, 111, 115
- system components
 - descriptions, 25
 - distribution, 31
- System Identity Synchronization for Windows. *See Identity Synchronization for Windows*, 21
- system manager
 - accepting certificates, 177
 - description, 28
- SystemManagerBootParams.cfg file, 209

T

- tabs
 - Configuration, 82, 83
 - Status, 82
 - Tasks, 82
- Tasks tab, 82
- TEMP directory, 144
- To Do list, 47, 76, 134, 144, 147
- To Do node, 189, 197
- troubleshooting, central logger, 191
- trusted certificates, 101, 171

U

- uid attribute, 155
- uninstall.cmd scripts, 162
- uninstallation failures, 207
- uninstalling
 - consoles, 166-167
 - Core, 161, 164
 - Directory Server Plugin, 161
 - Identity Synchronization for Windows, 161
 - software, 161
- UNIX commands
 - verifying Java Home, 74
- updates, detecting, 34

URLs

- Administration Server, 78
- configuration directory, 71, 139
- usage information, idsync, 206

user

- attributes, 52
- authentication failures, 39
- deletions, 128
- distinguished names, 96
- domain base DN, specifying, 131

user DNs

- example, 87, 96
- specifying, 87, 96

- User objectclass, 61

users

- adding to Active Directory, 55
- creating SULs, 52
- defining, 52
- filtering, 131, 230
- linking/synchronizing, 41, 49, 61, 64, 105, 207
- NT SAM domain, 153
- resynchronizing, 215
- special on Active Directory, 157
- subtrees, 41

using

- custom methods for Directory Server, 119
- SSL, 170, 181, 188

- USNchanged attribute, 35, 37

- UTF-8, 206, 221

utilities

- command line, 28
- forcepwchg, 221
- keytool, 177

V

validating

- certificates, 176, 177
- configurations, 134
- validation errors, 134

W

warnings, configuration, 134

Watchdog process, 26

WatchList.properties, 176

websites

- Directory Server publications, 59

- download Java Development Kit, 67

- Microsoft certificate authority, 59

- Microsoft product documentation, 59, 60

- Sun product documentation, 21

Windows

- configuring SSL, 59

- creating directory sources, 95

- removing Identity Synchronization for
Windows, 167

- running the installation program, 69

- selecting Directory Source, 130

- starting/stopping services, 83

Windows NT

- change detection, 36

- connector description, 29

- creating directory sources, 102

- enabling auditing, 199

- installing connectors, 148

- installing connectors and subcomponents, 33
Registry, 40

- specifying domain name, 103

- synchronization settings, 50

writing

- logs to files, 194

- logs to syslog daemon, 195

X

XML configuration documents

- importing exported 1.0 configurations, 77

- linking users, 64, 216

- linkusers.cfg, 224-226

- linkusers-simple.cfg, 223-224

- samples, 223

