



Identity Connect 2.1.0 Implementation Guide

Copyright © 2014-2016 salesforce.com. All rights reserved. Salesforce.com is a registered trademark of salesforce.com, Inc., as are other names and marks. Other marks appearing herein may be trademarks of their respective owners.

Abstract

Guide to installing and configuring Identity Connect.

Last updated : July 31, 2017

Table of Contents

Preface	v
1. Who Should Use this Guide	v
2. Formatting Conventions	v
1. Overview of an Identity Connect Deployment	1
1.1. Overview of the Identity Connect Architecture	1
1.2. Outline of the Setup Process	2
2. Getting Identity Connect Up and Running	4
2.1. Downloading, Installing, and Starting Identity Connect	4
2.2. Stopping and Restarting Identity Connect	5
2.3. Installing Identity Connect as a Windows Service	8
2.4. Running Identity Connect as a Service on UNIX-Like Systems	10
2.5. Updates to Your Salesforce Organization Required by Identity Connect	11
2.6. Upgrading an Identity Connect Instance	15
2.7. Using Identity Connect With the Salesforce1 Mobile App	17
3. Configuring Connections Between Identity Connect, Active Directory, and Salesforce	18
3.1. Configuring the Data Source	19
3.2. Configuring the Salesforce Connector	25
3.3. Connecting to More Than One Salesforce Organization	30
3.4. Delete a Salesforce Organization Configuration	31
3.5. General Notes About the User Interface	32
4. Mapping Data Between Active Directory and Salesforce	34
4.1. Mapping Attributes	35
4.2. Mapping Salesforce Profiles to Active Directory Groups	39
4.3. Mapping User Roles to Active Directory Groups	42
4.4. Mapping Permission Sets to Groups	43
4.5. Permission Set Licenses	45
4.6. Mapping Salesforce Groups to Active Directory Groups	46
5. Data Synchronization and User Association Management	48
5.1. Overview of the Synchronization Process	48
5.2. Managing User Associations	49
5.3. Configuring the Synchronization Schedule	54
5.4. Increasing the Number of Connections for Multiple Synchronizations	55
6. Configuring Single Sign-On	56
7. Configuring Identity Connect for Integrated Windows Authentication (Advanced Feature)	59
7.1. Before You Start	61
7.2. Configuring the Kerberos User and Creating the Keytab	61
7.3. Configuring the Authentication Filter in Identity Connect	65
7.4. Configuring Client Browsers for SPNEGO	68
8. Customizing the Identity Connect Interface	73
8.1. Customizing the UI Theme	73
8.2. Changing the Password Reset Link	73
8.3. Changing the Session Timeout	74

9. Configuring Auditing and Reporting	75
9.1. Running Reconciliation Reports	75
9.2. Running Synchronization Reports	76
9.3. Running User Activity Reports	80
10. Securing an Identity Connect Deployment	82
10.1. Managing SSL Certificates	82
10.2. Configuring Identity Connect for Client Certificate Authentication	89
10.3. Obfuscating Bootstrap Information	90
11. Installing an Alternative Repository	91
11.1. Setting Up Identity Connect With MySQL	91
11.2. Setting Up Identity Connect With MS SQL Server	94
12. Deploying Identity Connect for High Availability	100
12.1. Configuring High Availability With MySQL	100
12.2. Configuring a Load Balancer	102
12.3. Configuration Changes in a Clustered Environment	103
13. Advanced Configuration	104
13.1. Managing the Internal Repository	104
13.2. Working With Identity Connect Log Files	109
13.3. Using Identity Connect for Delegated Authentication	110
13.4. Synchronizing Passwords With the Active Directory Password Sync Plugin (Advanced Feature)	110
13.5. Managing Scheduled Tasks in Identity Connect	127
14. Troubleshooting an Identity Connect Installation	130
14.1. Troubleshooting the Integrated Windows Authentication Configuration	130
14.2. Recreating the Identity Connect Repository	138
14.3. General Troubleshooting	140
Identity Connect Glossary	141
Index	142

Preface

This guide shows you how to install, configure, and manage Identity Connect 2.1.0.

1. Who Should Use this Guide

This guide is written for administrators of Identity Connect and covers the install, configuration, and removal procedures that you theoretically perform only once per version. This guide also covers the configuration and management of the synchronization mechanism that ensures consistency across two disparate data stores.

The Identity Connect software is based on the OpenIDM and OpenAM products. For a deeper understanding of how the product works, you can have a look at the OpenIDM and OpenAM documentation, although such information is not required for basic installation, configuration, and management of Identity Connect.

2. Formatting Conventions

Most examples in the documentation are created in GNU/Linux or Mac OS X operating environments. If distinctions are necessary between operating environments, examples are labeled with the operating environment name in parentheses. To avoid repetition file system directory names are often given only in UNIX format as in `/path/to/server`, even if the text applies to `C:\path\to\server` as well.

Absolute path names usually begin with the placeholder `/path/to/`. This path might translate to `/opt/`, `C:\Program Files\`, or somewhere else on your system.

Command-line, terminal sessions are formatted as follows:

```
$ echo $JAVA_HOME
/path/to/jdk
```

Command output is sometimes formatted for narrower, more readable output even though formatting parameters are not shown in the command.

Program listings are formatted as follows:

```
class Test {
    public static void main(String [] args) {
        System.out.println("This is a program listing.");
    }
}
```

Chapter 1

Overview of an Identity Connect Deployment

Identity Connect enables you to upload user data from your enterprise data store (Active Directory) to one or more Salesforce organizations, and automatically to synchronize this data when user entries are added, changed, or removed. In addition, Identity Connect enables single sign-on (SSO) to Salesforce, using the Security Assertion Markup Language (SAML).

1.1. Overview of the Identity Connect Architecture

Identity Connect includes a browser-based user interface, and is installed “on premises”, inside your Network. A customizable UI wizard enables you to configure data synchronization from your Active Directory server to your Salesforce organization.

A single Active Directory server can be connected to multiple Identity Connect instances, each targeting a separate Salesforce organization. This enables you to synchronize a sandbox organization and a production organization from the same Active Directory server. In addition, you can connect one Identity Connect instance to multiple Salesforce organizations. For example, if two organizations merge, and the user data for both organizations is stored in a single Active Directory server, Identity Connect can synchronize that Directory Server data simultaneously to multiple Salesforce organizations.

When Identity Connect has been installed and configured, any access to the subdomain of your organization on Salesforce (such as [example.salesforce.com](#)) can be configured to go through Identity Connect. Attempts to access [example.salesforce.com](#) directly are rerouted to Identity Connect, which manages access.

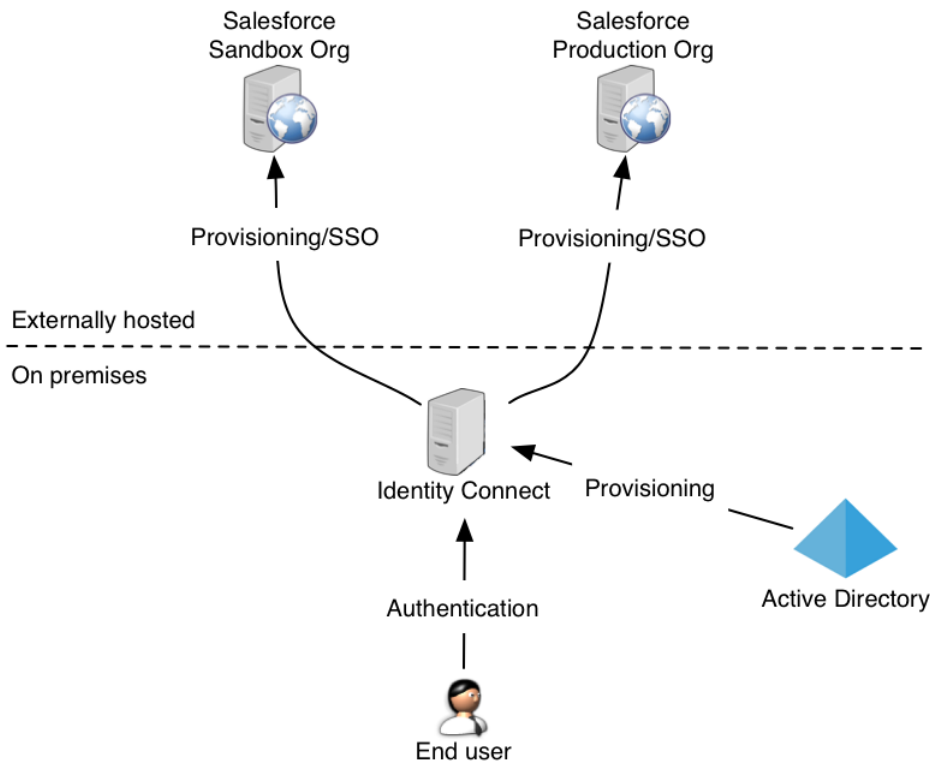
Although Identity Connect manages user data across disparate data stores, users and passwords are generally not stored in Identity Connect itself. Administrative access to Identity Connect relies on the credentials of administration users in Active Directory.

When an administrative user logs into Identity Connect (with the URL <https://hostname.domain:8443/admin/>), he is able to configure, manage and monitor data synchronization between Active Directory and Salesforce. If single sign-on has been configured, and the AD user has been linked to his Salesforce account, a regular user can log into Identity Connect (with the URL <https://hostname.domain:8443/connect/>), and is routed directly to his Salesforce dashboard, via SAML.

The session for the administration UI is shared with the user UI. Therefore, when an administrator is logged into Identity Connect, and logs into his Salesforce user login page, he does so without entering additional authentication details.

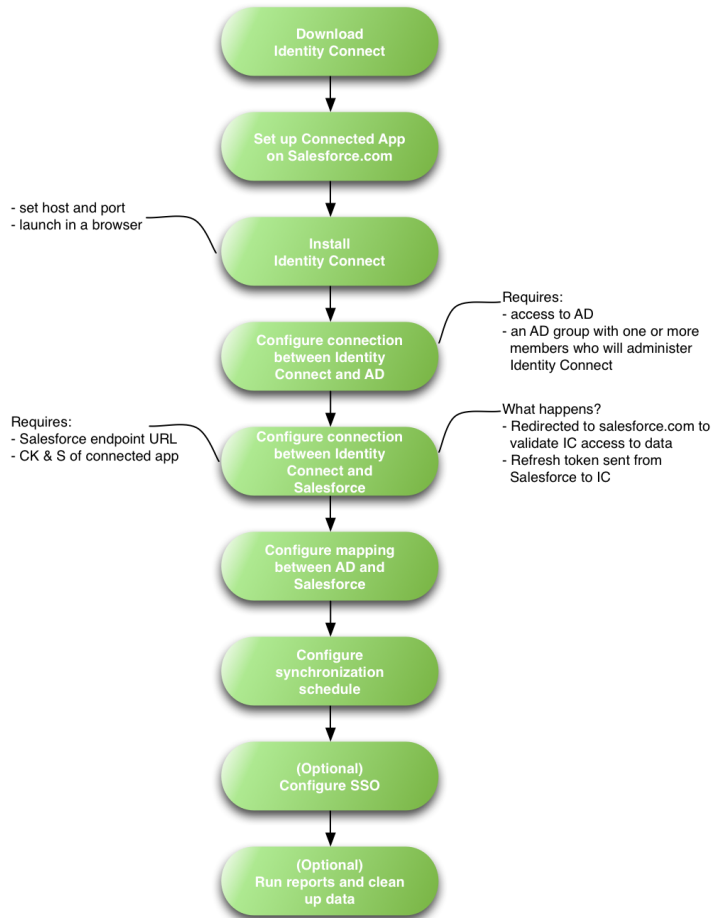
By default, access to Identity Connect is controlled with forms-based authentication. Users of Identity Connect provide the login credentials of their Active Directory account when they log in. You can configure Identity Connect for Integrated Windows Authentication (IWA) in addition to forms-based authentication. For more information, see Chapter 7, "*Configuring Identity Connect for Integrated Windows Authentication (Advanced Feature)*". You can also configure single sign-on (SSO) to Salesforce using the Security Assertion Markup Language (SAML). For more information, see Chapter 6, "*Configuring Single Sign-On*".

The following figure provides a high-level overview of the Identity Connect components, assuming two Salesforce organizations synchronized with a single Active Directory server.



1.2. Outline of the Setup Process

Setting up Identity Connect involves the configuration of multiple systems. The following flowchart provides a high-level overview of what happens between these systems during the setup process. Each step is discussed in more detail in the rest of this guide.



Chapter 2

Getting Identity Connect Up and Running

This chapter describes how to complete the initial configuration of an Identity Connect instance, and how to upgrade an existing instance to the latest Identity Connect version.

2.1. Downloading, Installing, and Starting Identity Connect

Download Identity Connect from the URL provided to you by your Salesforce representative, then use one of the following procedures to install Identity Connect, depending on your operating system.

Procedure 2.1. To Install Identity Connect on UNIX-Like Systems

1. Unpack the contents of the .zip file into the install location.

```
$ cd /path/to
$ unzip ~/Downloads/salesforce_identity_connect_linux.zip
```

2. Run the setup script.

```
$ cd /path/to/salesforceIdConnect
$ ./setup.sh
```

3. Enter the SSL port to listen on for the Identity Connect user interface. The default is 8443.

If you are running Windows Firewall, make sure that inbound connections to this port are not blocked.

4. Enter **y** to have the Identity Connect server start immediately after setup, and run in the background.

When Identity Connect runs in the background, any log messages are output to the file `/path/to/salesforceIdConnect/logs/console.out`.

If you select not to have the server start immediately, you must start Identity Connect manually, using the `startup.sh` script. In this case, log messages are output to the terminal in which you started Identity Connect. To redirect log messages to `console.out`, follow the instructions in Section 2.2, "Stopping and Restarting Identity Connect".

5. Point your browser to `https://hostname.domain:8443/admin/`, (specifying an alternate port if you entered an alternate port during the setup).

You will receive a warning about the website's security certificate if you have not replaced the default certificate with a trusted certificate. For more information, see Section 10.1, "Managing SSL Certificates".

Procedure 2.2. To Install Identity Connect on Windows Systems

1. Double-click the `salesforce_identity_connect_win.zip` file and select Extract all files.
2. In a command window, change to the `install-location/salesforceIdConnect` directory.

```
C:\>cd install-location\salesforceIdConnect
```

3. Before you start the setup, consider the HTTPS port on which Identity Connect should listen. By default, Identity Connect listens on port 8443. To specify a different port, edit the `openidm.port` `.https` property in the `conf/boot/boot.properties` file before you start Identity Connect.
4. Run the `startup.bat` script.

```
C:\install-location\salesforceIdConnect\>startup.bat
```

Messages are output to the Felix shell in the command window in which you launched Identity Connect.

5. Point your browser to `https://hostname.domain:8443/admin/`, (specifying an alternate port if you changed the default port).

You will receive a warning about the website's security certificate if you have not replaced the default certificate with a trusted certificate. For more information, see Section 10.1, "Managing SSL Certificates".

Note

The remainder of this Guide assumes that Identity Connect is accessed at the URL `identityconnect.example.com`. Replace `identityconnect.example.com` in all URLs pertaining to the UI with the FQDN of the host on which Identity Connect is installed.

If you have the "ADBlock" extension enabled for your browser, disable it. The "ADBlock" extension filters all pages that include "AD" which interferes with several Active Directory pages.

2.2. Stopping and Restarting Identity Connect

You can check whether an instance of Identity Connect is running, stop, and restart the server as outlined in the following sections.

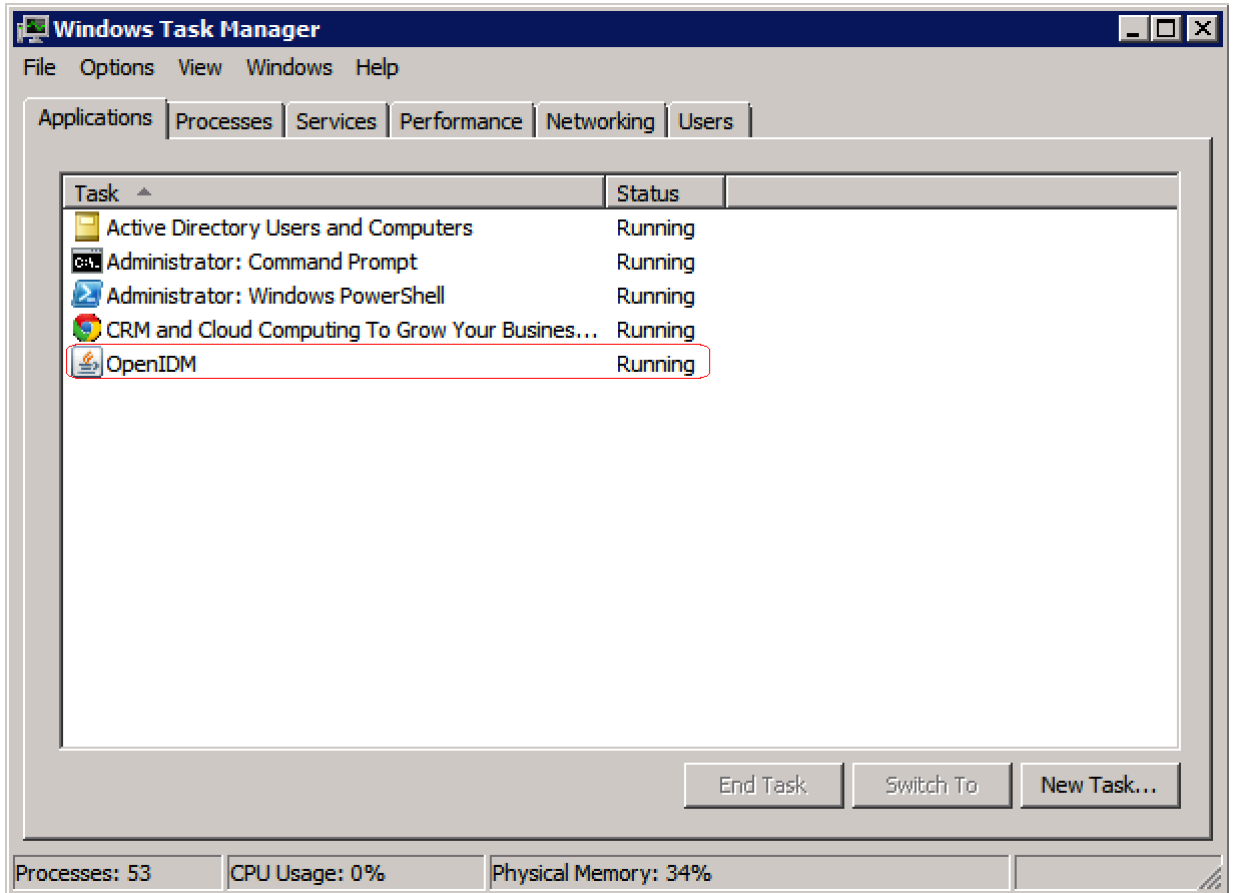
- To check whether Identity Connect is running on UNIX-like systems, run the following command on the system on which you started Identity Connect:

```
$ ps -ef | grep openidm
```

If an instance of Identity Connect is running, you should see output similar to the following:

```
501 91957      1  0  4:47PM ttys001  12:03.23 /usr/bin/  
java  
-Djava.util.logging.config.file=/path/to/salesforceIdConnect/conf/logging  
.properties  
-Xmx2048m -  
Xms2048m  
-Djava.endorsed.dirs= -classpath /path/to/salesforceIdConnect/bin/*:  
/path/to/salesforceIdConnect/framework/  
*  
-Dopenidm.system.server.root=/path/to/  
salesforceIdConnect  
-Djava.awt.headless=true org.forgerock.commons.launcher  
.Main  
-c bin/launcher.json
```

- To check whether Identity Connect is running on Windows systems, check the running applications in the Windows Task Manager. Identity Connect runs under the application "OpenIDM".



- To stop Identity Connect on UNIX-like systems, run the shutdown script, located in the install directory, or type **shutdown** in the Felix console that opened when you started Identity Connect.

```
$ cd /path/to/salesforceIdConnect
$ ./shutdown.sh
./shutdown.sh
Stopping OpenIDM (91957)
```

- To stop Identity Connect on Windows systems, stop the OpenIDM application in the Windows Task Manager, or type **shutdown** in the Felix console that opened when you started Identity Connect.
- To restart Identity Connect on UNIX-like systems, run the startup script, located in the install directory. Use the **nohup** command to keep Identity Connect running after you log out, and redirect the console output to **console.out**, as follows.

```
$ cd /path/to/salesforceIdConnect
$ nohup ./startup.sh > logs/console.out 2>&1&
[1] 32548
```

- To restart Identity Connect on Windows systems, run the `startup.bat` script in the install directory.

2.3. Installing Identity Connect as a Windows Service

You can install Identity Connect to run as a Windows service, so that the server starts and stops automatically when Windows starts and stops. You must be logged in as an administrator to install Identity Connect as a Windows service.

Note

On a 64-bit Windows server, you must have a 64-bit Java version installed to start the service. If a 32-bit Java version is installed, you will be able to install Identity Connect as a service, but starting the service will fail.

Before you launch the `install-service.bat` file, which registers the `IdentityConnect` service within the Windows registry, make sure that your `JAVA_HOME` environment variable points to a valid 64-bit version of the JRE or JDK. If you have already installed the service with the `JAVA_HOME` environment variable pointing to a 32-bit JRE or JDK, delete the service first, by running `sc delete IdentityConnect` from a Windows command prompt, then re-install the service.

1. Unpack the Identity Connect .zip file, as described previously, and change to the `bin` directory:

```
C:\>cd install-location\salesforceIdConnect\bin
```

2. Launch the Identity Connect service, with the following command:

```
C:\install-location\salesforceIdConnect\bin>install-service.bat
Identity Connect Service successfully installed as "IdentityConnect" service
```

3. Use the Windows Service manager to manage the Identity Connect service.

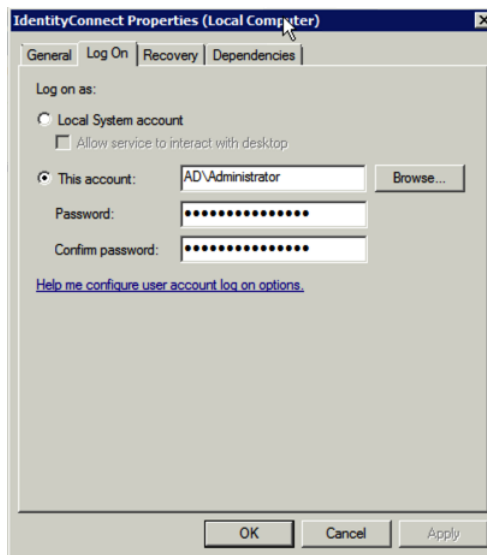
The screenshot shows the Windows Services (Local) console. The 'IdentityConnect' service is highlighted with a red box. The table below represents the data visible in the screenshot.

Name	Description	Status	Startup Type	Log On As
Disk Defragmenter	Provides Disk Defragm...		Manual	Local System
Distributed Link Tra...	Maintains links between...		Manual	Local System
Distributed Transac...	Coordinates transactio...	Started	Automatic (D...	Network S...
DNS Client	The DNS Client service...	Started	Automatic	Network S...
DNS Server	Enables DNS clients to ...	Started	Automatic	Local System
Encrypting File Syst...	Provides the core file ...		Manual	Local System
Extensible Authenti...	The Extensible Authen...		Manual	Local System
File Replication Ser...	Synchronizes folders ...	Started	Automatic	Local System
Function Discovery ...	The FDPHOST service ...		Manual	Local Service
Function Discovery ...	Publishes this compute...		Manual	Local Service
Group Policy Client	The service is responsi...	Started	Automatic	Local System
Health Key and Cer...	Provides X.509 certific...		Manual	Local System
Human Interface D...	Enables generic input ...		Manual	Local System
IdentityConnect	Provides Active Direct...		Automatic	Local System
IKE and AuthIPs...	The IKEEXT service ho...		Manual	Local System

4. Change the user account for this service from the default (`local system`) account to an account with administrative privileges. The `local system` account has limited permissions and an Identity Connect service that runs with this account will encounter problems during synchronization.

To change the user account:

- Double click the "IdentityConnect" service in the Windows Service manager.
- Select the Log On tab.
- Select This Account and browse for an Active Directory administrative account.
- Enter the password for the administrative account.



- Click Apply to save the changes.

5. Use the Windows Service Manager to start, stop, or restart the service.

To uninstall the Identity Connect service, run the following command:

```
C:\install-location\salesforceIdConnect\bin>launcher.bat /uninstall
Service "IdentityConnect" removed successfully
```

2.4. Running Identity Connect as a Service on UNIX-Like Systems

Identity Connect provides an RC script that generates an initialization script to run Identity Connect as a service on UNIX-like systems. You must start the initialization script manually, or automatically at boot time.

When Identity Connect runs as a service, logs are written to the directory in which Identity Connect was installed.

To run Identity Connect as a UNIX service:

1. If you have not already done so, install and set up Identity Connect, as described in Procedure 2.1, "To Install Identity Connect on UNIX-Like Systems".
2. Run the RC script.

```
$ cd /path/to/salesforceIdConnect/bin
$ ./create-idconnect-rc.sh
idconnect script has been created in /path/to/salesforceIdConnect/bin
To finish installation, copy the idconnect script into the /etc/init.d folder
and run the following command:
chkconfig --add idconnect
```

```
To remove the service, run the following command:
chkconfig --del idconnect
```

3. As a user with root privileges, copy the `idconnect` script to the `/etc/init.d` folder.

```
$ sudo cp idconnect /etc/init.d/
```

4. As a user with root privileges, run the `chkconfig --add` command to install the Identity Connect service.

```
$ sudo cd /etc/init.d/
$ sudo chkconfig --add idconnect
```

5. Start the Identity Connect service.

```
$ service idconnect start
```

Alternatively, reboot the system to start the Identity Connect service automatically.

6. (Optional) To stop, restart, or check the status of the service, use the following commands:

```
$ service idconnect stop
```

```
$ service idconnect restart
```

```
$ service idconnect status
```

7. (Optional) To remove the service, run the following command, as a user with root privileges:

```
$ sudo chkconfig --del idconnect
```

Note that this command does not remove the Identity Connect application, but prevents it from being run as a service.

2.5. Updates to Your Salesforce Organization Required by Identity Connect

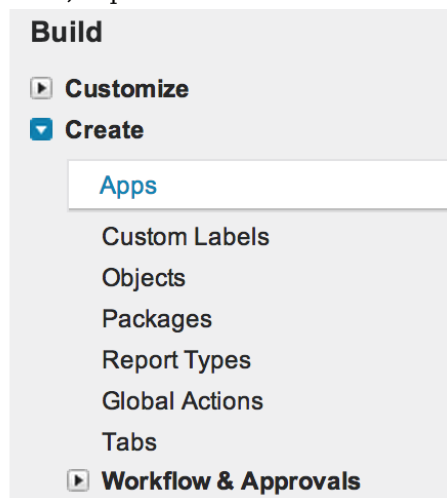
Identity Connect requires a few updates to your Salesforce organization configuration. To ensure an optimal installation and configuration experience, you should complete these Salesforce organization updates before you start configuring Identity Connect.

2.5.1. Setting Up a Connected App for Identity Connect

Identity Connect requires a Connected App to connect to salesforce.com using the OAuth 2.0 protocol. For more information about Connected Apps, see the [Connected Apps Overview](#) in the Salesforce documentation.

To set up a Connected App for Identity Connect, follow these steps:

1. Log in to salesforce.com with your Salesforce credentials.
2. Click *Setup* in the top right corner.
3. In the left hand menu, under *Build*, expand the *Create* item and click *Apps*.



4. On the right hand panel, scroll down to *Connected Apps* and click *New*.

Connected Apps		New
Connected App Name	Description	

- In the *New Connected App* panel, enter the following Basic Information:
 - Connected App Name.** Enter a name that you will recognize as the Identity Connect App, for example, `IdentityConnect`.
 - API Name.** Enter the application API name. Note that the Application API Name can only contain underscores and alphanumeric characters. The name must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
 - Contact Email.** Enter the email address of the person responsible for this Connected App within your organization, for example, `admin@example.com`.
- Select *Enable OAuth Settings* and enter the following information:
 - Callback URL.** Enter the Identity Connect URL, to which the requested token will be sent. The default callback URL is, `https://hostname.domain:8443/admin/index.html#salesforceCallback`.
 - Selected OAuth Scopes.** Click the *Add* button to add the following *Available Auth Scopes* to the *Selected OAuth Scopes* column:
 - Access and manage your data
 - Access your basic information
 - Perform requests on your behalf at any time

2.5.2. Registering a Domain in Salesforce

If you plan to configure single sign-on with Identity Connect, you must have a domain registered in Salesforce.

To register a domain, follow these steps:

1. Log in to Salesforce.com and navigate to your Salesforce Setup page.
2. From the Administer menu, expand the Domain Management item and select My Domain.
3. Enter a subdomain name for your domain.

Click Check Availability to make sure that the name is unique, then agree to the Terms and Conditions, and click Register Domain.

My Domain

Add a domain name to the URLs you use to log into and navigate Salesforce.com. Your domain name not only gives you another way to feature your brand, but also make your data more secure. Rollout is easy: after registering for your domain name, you can test it before opening it up to users. Built-in redirection tools help make the rollout seamless.

```
graph LR; A[Choose Domain Name] --> B[Domain Registration Pending]; B --> C((L)); C --> D[Domain Ready for Testing]; D --> E[Domain Deployed to Users];
```

First, choose a subdomain to register for your organization. Choose carefully, because you can only register a subdomain once for your organization.

Subdomain names can include up to 40 letters, numbers, or hyphens. Your subdomain can't start or end with a hyphen.

https:// -my.salesforce.com/ ✔ Available

I agree to the [Terms and Conditions](#)

It might take several minutes for the domain registration to be processed.

4. After the domain has been registered, test the URL (click the Click here to login button) and then click Deploy to Users.

2.6. Upgrading an Identity Connect Instance

Identity Connect 2.1.0 provides an upgrade mechanism that enables you to patch an existing configuration.

Before you upgrade, note the following requirements:

Upgrading in a Clustered Environment

In a clustered environment, you must shut down all nodes before applying the patch. Patch the `clustered-first` node first, then patch each `clustered-additional` node. When all nodes have been patched, restart each node in the order in which they were patched, that is, restart the `clustered-first` node first, then restart each `clustered-additional` node.

As indicated in the upgrade procedure, do not access the Administration console, or restart any subsequent instances in the cluster, until you have observed the `Completed post-upgrade tasks...` message. For more information about running Identity Connect in a clustered environment, see Chapter 12, "Deploying Identity Connect for High Availability".

Behavior of Permission Set Mappings

In Identity Connect prior to version 1.0.3, *all* permission sets within a Salesforce organization were synchronized to, and managed by, Identity Connect. This implied that explicit permission set assignments from within the Salesforce organization were not guaranteed to persist, and would eventually be removed.

From Identity Connect version 1.0.3 onwards, any permission sets that are not included in the permission set to Active Directory Group mapping page are excluded from the scope of what is managed by Identity Connect. These permission set assignments are therefore not added, or removed by Identity Connect. Note that if a permission set *is* included on the permission set to Active Directory Group mapping page, but is mapped to `None`, Identity Connect will effectively overwrite any explicit assignments from within the Salesforce organization for that permission set.

Upgrading from Identity Connect 1.0.2 to a newer version retains the behavior with regard to permission set mappings that was present in Identity Connect 1.0.2. This means that after an upgrade from 1.0.2, *all* permission sets are included in the mapping page and are managed by Identity Connect. If you want to control explicit permission set assignments after an upgrade from 1.0.2, you *must* remove the permission sets from the Identity Connect mapping page that you want to manage explicitly.

To upgrade Identity Connect to version 2.1.0, follow these steps:

1. Stop Identity Connect, if it is running.

```
$ cd /path/to/salesforceIdConnect
$ ./shutdown.sh
Stopping OpenIDM (81491)
```

2. Back up your existing configuration by zipping up the entire `salesforceIdConnect` directory, and the database, in the event that you are using an external repository.

There is currently no way to revert a patch, so it is highly recommended that you back up your configuration and data before patching. Although the upgrade process does create an archive of the current configuration, this information is not sufficient to revert an upgraded installation.

3. Download and unzip the Identity Connect patch (`salesforce_identity_connect_win_patch.zip` for Windows systems and `salesforce_identity_connect_linux_patch.zip` for Linux systems).
4. Run the following command to apply the patch to your existing Identity Connect instance.

```
$ cd ~/Downloads
$ java -jar salesforceIdConnect-2.1.0-patch.jar /path/to/salesforceIdConnect
Downloaded to salesforceIdConnect/patch/bin/salesforceIdConnect-2.1.0-patch.jar
Apr 07, 2014 4:42:51 PM org.forgerock.patch.Archive initialize
INFO: Created patch archive directory: salesforceIdConnect/patch/archive/20140407_164251
Apr 07, 2014 4:42:51 PM HistoryLog INFO: Applying "Salesforce Identity Connect Patch",
version=188.16
Apr 07, 2014 4:42:51 PM HistoryLog INFO: Target: salesforceIdConnect,
Source: file:/path/to/salesforceIdConnect-2.1.0-patch.jar
Apr 07, 2014 4:43:02 PM HistoryLog INFO: Completed
```

5. If you are using an external MySQL repository, import the data definition language script for the Identity Connect upgrade into MySQL.

```
$ cd /path/to/mysql
$ ./bin/mysql -u root -p < \
/path/to/salesforceIdConnect/db/scripts/mysql/upgrade-MySQL-schema.sql
Enter password:
$
```

Enter the root user password for the MySQL server.

6. If you are running Identity Connect as a Windows service, uninstall and reinstall the `IdentityConnect` service so that the appropriate changes are applied to the JVM startup parameters. To uninstall and reinstall the service, run these commands after the upgrade:

```
C:\install-location\salesforceIdConnect\bin>launcher.bat /uninstall
Service "IdentityConnect" removed successfully
C:\install-location\salesforceIdConnect\bin>install-service.bat
Identity Connect Service successfully installed as "IdentityConnect" service
```

7. Restart Identity Connect.

```
$ cd /path/to/salesforceIdConnect
$ nohup ./startup.sh > logs/console.out 2>&1&
[1] 32548
```

Caution

Do not access the Administration console until you have observed the `Completed post-upgrade tasks.` message in the OpenIDM log file on startup. This is applicable to all deployments, but is particularly important in a clustered environment because the first patched instance (`clustered-first`) patches a number of configuration objects, including synchronization mappings and managed objects. These configuration patches are required *before* any subsequent nodes are started, to ensure a cohesive configuration across the cluster.

8. Before logging into Identity Connect, clear your browser cache. The browser cache contains files from the previous Identity Connect release, that might not be refreshed when you log into the UI of the new release.

2.7. Using Identity Connect With the Salesforce1 Mobile App

There are certain specific requirements regarding the use of Identity Connect with the Salesforce1 Mobile App. This section lists these requirements.

Replace the Default SSL Certificate on the Identity Connect Host

As an Identity Connect administrator, you *must* deploy an SSL certificate on your Identity Connect host that is trusted by the mobile devices of your users.

Mobile applications will not work with the default self-signed certificate that is provided with Identity Connect. For more information, see Section 10.1, "Managing SSL Certificates".

Provide the Domain Name to the App

Your Salesforce1 Mobile App users must specify the correct domain for Identity Connect within their App.

Click on the gear icon at the top right of the App, and click the plus icon (+) to specify the connection details. Enter the host that corresponds to your Identity Connect instance, for example, `https://identityconnect.example.com:8443`. This must be the same URL that you specified during the Identity Connect setup (see Chapter 3, "Configuring Connections Between Identity Connect, Active Directory, and Salesforce").

Note

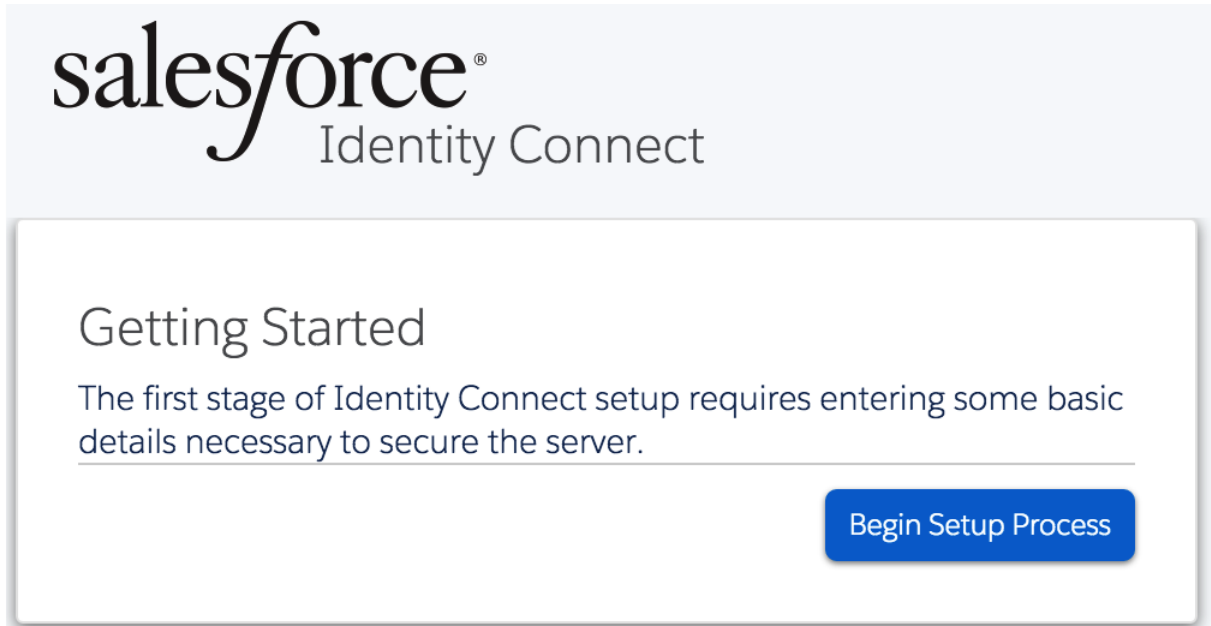
If you have configured IWA, but a user's mobile device does not support Kerberos, the Identity Connect login page on the Salesforce1 App will fall back to their form-based Active Directory login.

Chapter 3

Configuring Connections Between Identity Connect, Active Directory, and Salesforce

Part of the Identity Connect setup involves defining connections between Identity Connect and Active Directory, and between Identity Connect and Salesforce. This chapter describes how to configure these connections.

After you have set up Identity Connect and pointed your browser to <https://hostname.domain:8443/admin/>, the Identity Connect Getting Started page is displayed.

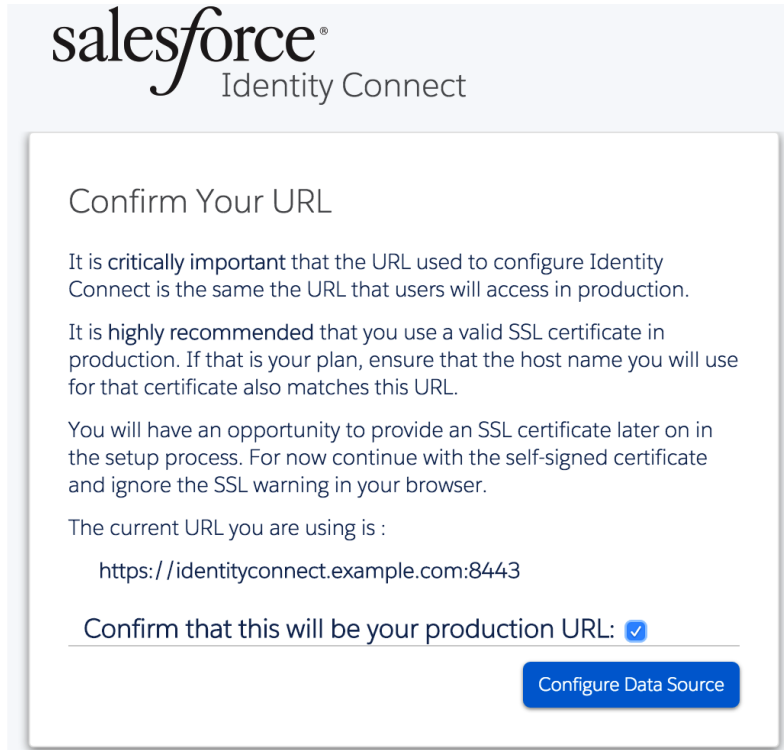


Click Begin Setup Process to start the setup.

A message requesting you to confirm the Identity Connect URL is displayed. The URL displayed here is the one that you are using to access the Identity Connect setup. This *must* be the same URL as the URL with which your users will access Identity Connect. If it is not the same URL, your SAML configuration will ultimately fail. For example, if you are configuring Identity Connect using the URL <https://localhost.com:8443>, but your users will ultimately be accessing Identity Connect at <https://>

`connect.example.com:8443`, the URL that is configured with SAML will not match the URL your users are using, and they will therefore be unable to log in with SAML.

If you realize at this point that this is not the URL with which your users will be accessing Identity Connect, cancel the setup, then access Identity Connect using the correct URL.



Select "Confirm that this will be your production URL" and click Configure Data Source to continue.

3.1. Configuring the Data Source

The first step in setting up Identity Connect is configuring the data source, or Active Directory connector. Identity Connect supports connections to a full Active Directory server, and to an ADLDS (Active Directory Lightweight Directory Services) instance.

3.1.1. Configuring the Active Directory Connector

1. Click *Configure Data Source* to configure the Active Directory connector.

2. On the *Data Source: Active Directory* page, provide the following information:

- Select the type of data source that is used as your data store (either Active Directory or Active Directory Lightweight Directory Services).
- *Host name or IP*. Enter the fully qualified host name, or IP address, of the machine that hosts the Active Directory instance.
- *Port*. Enter the port number on which the Active Directory server listens for LDAP connections. The default LDAP port is 389. The default LDAPS port is 636. If you are connecting to a Global Catalog, the default port is 3268, or 3269 if you are using SSL.

Check *Use SSL* to connect to the LDAPS port.

Make sure that remote LDAP or LDAPS access to the Active Directory server is allowed through the Firewall.

If you select to use SSL and the root CA for your Active Directory certificate is not in the trust store, you must provide the public SSL certificate for your Active Directory server as follows:

- On your Active Directory server, type the following command into a Command Prompt window:

```
C:\>certutil -ca.cert client.crt
```

This command will output the certificate (from `-----BEGIN CERTIFICATE-----` to `-----END CERTIFICATE-----`) to the command line.

- Copy the contents of the certificate to the clipboard.
- On the Identity Connect Active Directory configuration screen, click *show certificate*.
- Paste the certificate contents into the SSL Certificate window and click the close icon.
- *Account Distinguished Name (DN)*. Enter the bind DN of a user that will be used by the Active Directory connector to access Active Directory. This user must adhere to the following requirements:
 - Has at least read access to all of the base contexts that will be managed by Identity Connect
 - Is included in these base contexts
 - Is *not* filtered out by the user or group filters that you specify on this screen

In this example, the sample user, `Babs Jensen`, is used.

If Identity Connect is connecting to a single domain controller (DC), the user that is specified here must either be an administrative user (that is, a member of the `Administrator` group or the `Domain Admins` group) or a regular user that has been given the appropriate permissions.

A regular user generally does not have permission to access the `cn=Deleted Objects` container, and as a result, liveSync will have problems synchronizing deletions. If you specify a regular user here, you must grant the user `List Content` and `Read Properties` permissions on the `cn=Deleted Objects` container of that domain. To change user permissions, use the `dsacls` utility, as described in the Microsoft technet article at <http://technet.microsoft.com/en-us/library/cc771151.aspx>.

If Identity Connect is connecting to a Global Catalog (GC), liveSync does not synchronize deletions. For more information, see Section 3.1.2, "Working with Multiple Active Directory Domains".

- *Password*. Enter the bind password for the user specified in the previous step.

The bind DN and password of this Active Directory connector user are stored in the connector configuration file named `/path/to/salesforceIdConnect/conf/provisioner.openicf-ldap.json`. If you change the bind password of the Active Directory connector user in Active Directory, the connection from Identity Connect to Active Directory will fail because the connection credentials will be invalid. Therefore, a password change for this user, in Active Directory, must also be made in the Identity Connect configuration.

To change the connection user password in Identity Connect, edit the `/path/to/salesforceIdConnect/conf/provisioner.openicf-ldap.json` file, updating the `"credentials"` configuration property, for example:

```
"configurationProperties" : {  
  "passwordAttributeToSynchronize" : null,  
  ...  
  "credentials" : "NewPassw0rd",  
  ...  
}
```

The connection user password is encrypted as soon as the file is saved.

- *Attribute used for login.* Select the attribute with which users will log in to the Identity Connect user interface.

By default, for a full Active Directory Server, the login attribute is `sAMAccountName`. For an AD LDS instance, the login attribute is `displayName`. However, you can select any attribute here. Be sure that the attribute you select will have a unique value for each user.

- *Base Contexts.* Enter the path to one or more base DN's that will be synchronized during the data synchronization phase.

Note

Make sure that the user and group entries that will be managed through Identity Connect are included in the base contexts that you specify here.

- *User Filter.* Specify one or more LDAP filters that will be applied to the Active Directory users, to determine which users will be mapped to Salesforce accounts.

By default, Identity Connect filters out computer entries, with the filter `(!(objectClass=Computer))`.

Click the User Filter field and use the Update User Filter dialog to create additional filters, or to specify that no user filter should be applied.

For information about LDAP filter syntax, see <http://social.technet.microsoft.com/wiki/contents/articles/5392.active-directory-ldap-syntax-filters.aspx>.

Update User Filter

((!(objectClass=Computer))

None of...

The value for

objectClass Matching Computer

Save

- Provide one or more Active Directory object classes to search for Identity Connect user entries. The default object class for user entries is **user**.
- *Group Filter*. Specify one or more LDAP filters that will be applied to Active Directory group entries, to determine which groups will be mapped to Salesforce profiles.

By default, Identity Connect filters out entries under the organizational unit **cn=Domain Users**, with the filter **((!(cn=Domain Users))**. **Domain Users** is a special group that typically includes *all* user entries in the directory, but is not displayed under a user's **memberOf** attribute (so the group displays no members when it is searched). Do not remove this filter from the configuration.

Click the Group Filter field and use the Update Group Filter dialog to create additional filters.

For information about LDAP filter syntax, see <http://social.technet.microsoft.com/wiki/contents/articles/5392.active-directory-ldap-syntax-filters.aspx>.

- Provide one or more Active Directory object classes to search for Identity Connect group entries. The default object class for group entries is **group**.
3. When you have completed all of the preceding fields, click *Validate Settings* to validate the data source configuration. If the configuration is valid, a validation message is displayed at the top of the page. If the configuration is not valid, a validation error is displayed, with additional details provided at the bottom of the page.

When the data source configuration is valid an Active Directory connector is created on the Identity Connect machine to facilitate access between Identity Connect and Active Directory.

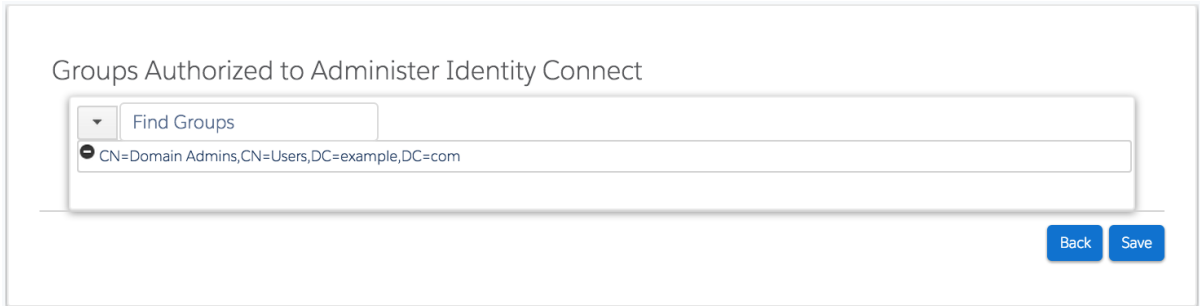
4. The following page lists all the defined groups on the Active Directory server, for the base context and object classes that were specified in the previous step.

It might take a couple of seconds for this list to be populated, depending on your network latency, as Identity Connect accesses the Active Directory here and reads the list of defined groups.

Select the group or groups whose members will be granted administration privileges for Identity Connect and click *Save*.

Caution

If you select a group here to which your own administrative account does not belong, you will be locked out of the Identity Connect administrative interface immediately and will be required to reinstall Identity Connect from scratch. You should therefore take care when selecting these groups.



The Active Directory connector has now been configured. At this point, the user interface exits and you are forced to authenticate (using the credentials established previously) before you can proceed with the configuration.

3.1.2. Working with Multiple Active Directory Domains

If your directory service has only one domain controller, or if all your Salesforce users are in the same domain, Identity Connect can connect to a single domain controller. If your directory service spans multiple domains, Identity Connect must connect to the Global Catalog (GC) to have a comprehensive view of all the domains. Multiple connections to multiple Domain Controllers from a single Identity Connect instance are not supported.

Using a GC as the authoritative data source has the following limitations:

- Only a subset of attributes is replicated from other domains to the GC.

Certain required attributes might be missing for the purposes of Identity Connect. To avoid this problem, you must modify the Active Directory schema to ensure that the required attributes are replicated to the GC. For more information, see Section 3.1.3, "Updating the Active Directory Schema for a Global Catalog".

- Delete operations are not detected immediately.

A liveSync operation will therefore not update the Salesforce data store with the result of a delete operation. Delete operations are detected by a reconciliation operation, so data stores are only temporarily "out of sync" with regard to deletes.

- Not all group types are supported.

Group membership information is replicated to the GC for *universal* groups only. You must therefore use universal groups if your directory service has more than one domain.

3.1.3. Updating the Active Directory Schema for a Global Catalog

To ensure that the attributes required by Identity Connect are replicated to the GC, you must update the Active Directory schema to include the required attributes. Before you update the schema, note the following:

- Only a member of the Schema Admins group can modify the Active Directory schema.
- Modifying the Active Directory schema requires a change to the registry on the Schema Master. For information about how to change the registry, see the Microsoft Knowledge Base article on *Registry Modification Required to Allow Writing to Schema*.

Modifying the registry incorrectly can severely compromise your system so exercise caution.

If you attempt to change the schema before you change the registry key, Active Directory will reject the change.

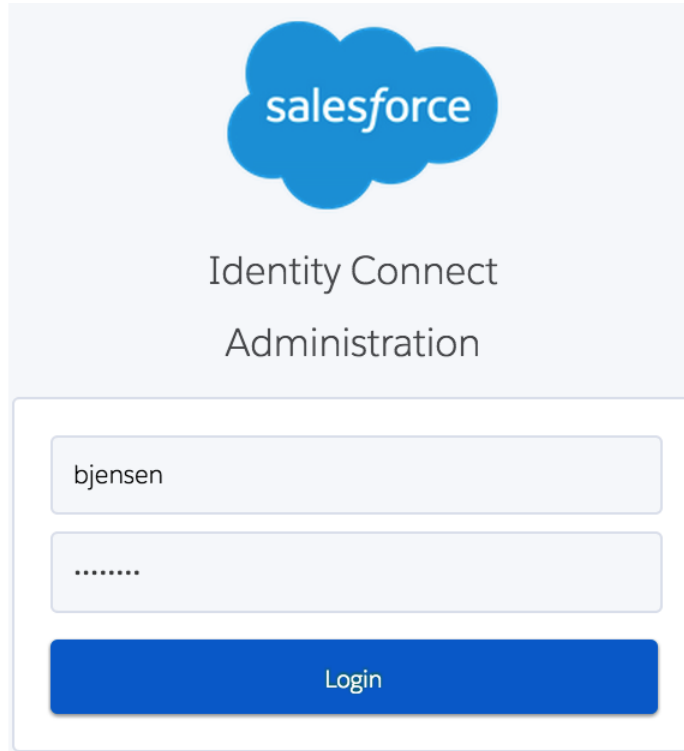
- Increasing the number of attributes that are replicated to the GC will invariably have an impact on network replication traffic.

Use the Active Directory Schema Microsoft Management Console (MMC) to modify the schema. For more information, see the Microsoft Knowledge Base article on *Modifying Attributes That Replicate to the Global Catalog*.

3.2. Configuring the Salesforce Connector

Identity Connect supports the configuration of multiple Salesforce organizations for a single Active Directory server. This enables you to synchronize two separate Salesforce organizations with the same Active Directory user data. When you configure the Salesforce Connector, you are prompted to select the Salesforce organization that will be the target data store for this specific connector. When you are configuring Identity Connect for the first time, you can only connect to a *new* Salesforce Organization. For subsequent configurations, a list of previously configured Salesforce organizations is provided. For more information on working with multiple Salesforce organizations, see Section 3.3, "Connecting to More Than One Salesforce Organization".

After you have completed the Active Directory connector configuration, the administration login page is displayed.



salesforce

Identity Connect
Administration

bjensen

.....

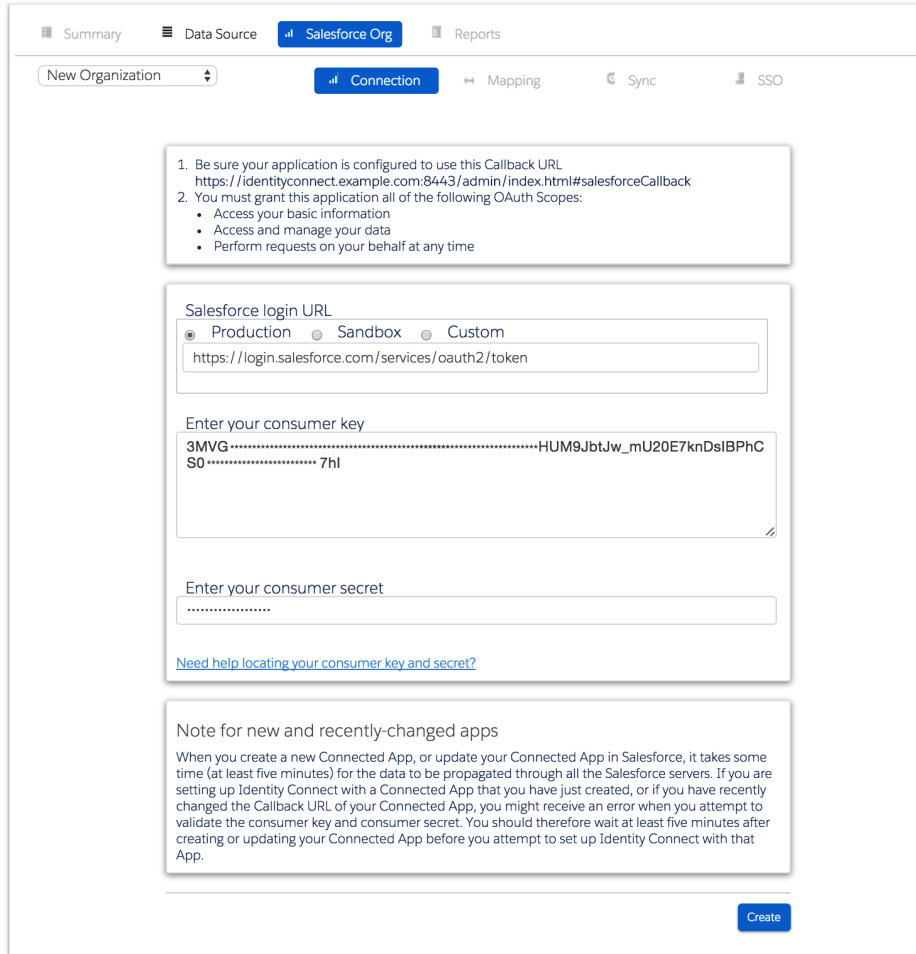
Login

Configure the Salesforce connector as follows:

1. Log in with the credentials of a user who belongs to one of the groups that you specified in the previous step.

If your data source is an AD LDS instance, you must log in with the `displayName` of a user in one of these groups.

The Salesforce connector setup page is displayed.



Summary | Data Source | **Salesforce Org** | Reports

New Organization | **Connection** | Mapping | Sync | SSO

1. Be sure your application is configured to use this Callback URL
`https://identityconnect.example.com:8443/admin/index.html#salesforceCallback`
2. You must grant this application all of the following OAuth Scopes:
 - Access your basic information
 - Access and manage your data
 - Perform requests on your behalf at any time

Salesforce login URL

Production Sandbox Custom

`https://login.salesforce.com/services/oauth2/token`

Enter your consumer key

3MVG.....HUM9JbtJw_mU20E7knDsIBPhC
S0.....7hl

Enter your consumer secret

.....

[Need help locating your consumer key and secret?](#)

Note for new and recently-changed apps

When you create a new Connected App, or update your Connected App in Salesforce, it takes some time (at least five minutes) for the data to be propagated through all the Salesforce servers. If you are setting up Identity Connect with a Connected App that you have just created, or if you have recently changed the Callback URL of your Connected App, you might receive an error when you attempt to validate the consumer key and consumer secret. You should therefore wait at least five minutes after creating or updating your Connected App before you attempt to set up Identity Connect with that App.

Create

2. If this is the first time you are configuring Identity Connect, select *New Organization* (the only option available from the dropdown menu).
3. In the Salesforce login URL field, specify the OAuth endpoint that will be used to make the OAuth authentication request to Salesforce.
 - Select *Production* for a production system. The default endpoint for a production system is `https://login.salesforce.com/services/oauth2/token`.
 - Select *Sandbox* if you are verifying authentication on a test or sandbox organization. The default sandbox endpoint is `https://test.salesforce.com/services/oauth2/token`.
 - Select *Custom* to provide your own login URL.

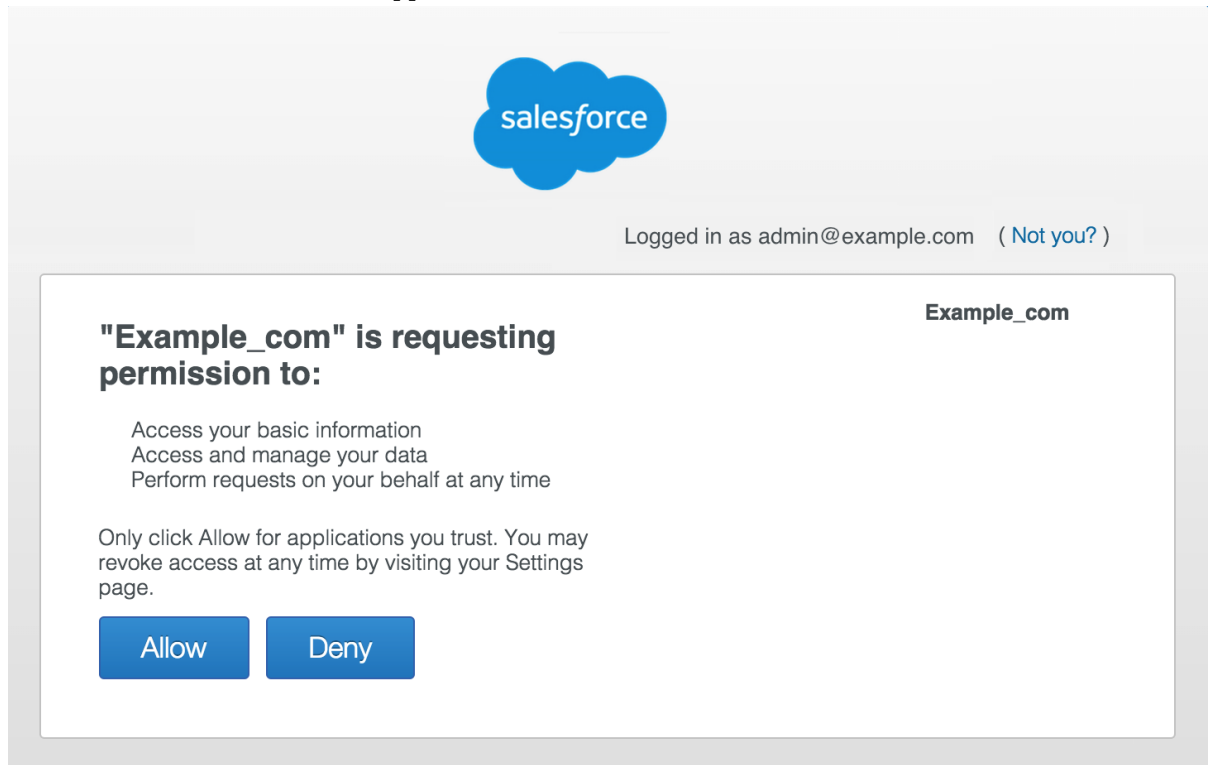
4. Enter your consumer key and consumer secret, acquired during the configuration of the Identity Connect Connected App (see Section 2.5.1, "Setting Up a Connected App for Identity Connect").

Make sure that the Callback URL specified on this screen is the one that you used when you set up the Identity Connect Connected App (<https://hostname.domain:8443/admin/index.html#salesforceCallback> by default).

Click Update to validate the consumer key and secret with Salesforce.

When you create a new Connected App, or update your Connected App in Salesforce, it takes some time (at least five minutes) for the data to be propagated through all the Salesforce servers. If you are setting up Identity Connect with a Connected App that you have just created, or if you have recently changed the Callback URL of your Connected App, you might receive an error when you attempt to validate the consumer key and consumer secret. Wait at least five minutes after creating or updating your Connected App before you attempt to set up Identity Connect with that App.

5. You are redirected to the Salesforce login page. Log in with your Salesforce credentials. Click *Allow* to authorize the remote application.

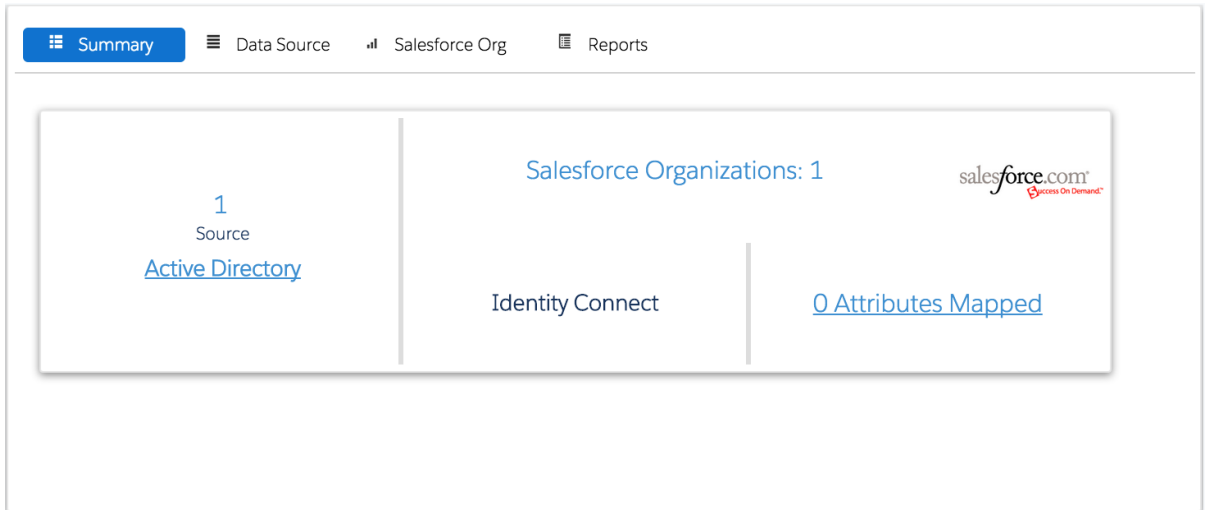


- You are redirected to Identity Connect. A confirmation message is displayed: "Successfully retrieved token from Salesforce!".

On networks with very high latency, the connection to Salesforce might timeout. In this case, you will see the following message: **Establishing Salesforce connection is taking longer than expected.**

This issue is generally resolved by clicking Test Connection to reestablish the connection.

- The Summary page is displayed, indicating your data source (Active Directory), the Salesforce organization that has been connected, and the status of any mapping that has been configured. At this stage, nothing has been mapped, so the mapping status indicates "0 Attributes Mapped".



You are now ready to move on to the mapping configuration.

Warning

If you have previously configured Identity Connect for your Salesforce organization, and you specified a custom login URL, you are prompted to use that same Identity Connect instance when you use the same custom URL on the Salesforce connector page. If the original Identity Connect instance was removed (and is being replaced) the new installation can result in an infinite loop as the validation attempts to locate the original instance.

In this case, you must either use the production URL or change your Salesforce organization configuration so that it does not use the Identity Provider for login.

When you refresh a Salesforce sandbox instance, your organization ID changes. As a result, the Identity Connect instance that has been configured for that organization then has an incorrect organization ID. Subsequent to the refresh, you will therefore see a connection error as Identity Connect attempts to connect to the old organization ID.

The easiest way to restore functionality, with the correct organization ID, is to delete the Salesforce connector, and recreate it. To delete the connector, select the Salesforce Org tab, then select Connection and click Delete. You can then recreate the connector. Remember that your new sandbox instance must have an active domain

configured. When you recreate the Salesforce connection, all previous configuration in the Mappings page, as well as the synchronization reports, are lost.

3.3. Connecting to More Than One Salesforce Organization

You might want to connect a single Active Directory instance to more than one Salesforce organization. For example, if two organizations merge, and the user data for both organizations is stored in a single Active Directory server, you can use a single Identity Connect instance to synchronize those multiple organizations simultaneously.

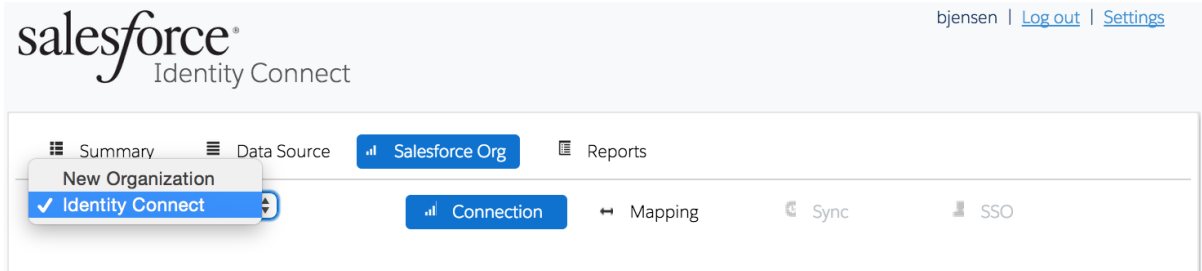
Caution

If you are setting up connections in a *test or sandbox* environment, and in a *production* environment, do not use this multiple organization feature. Rather set up separate Identity Connect instances for the test organization and the production organization.

1. Before you start, make sure that you have configured a Connected App for the new organization, as described in Section 2.5.1, "Setting Up a Connected App for Identity Connect".
2. To connect to an additional Salesforce organization, click **Salesforce Org** on the Summary page.

The dropdown list displays any previous organizations to which you have connected with this Identity Connect instance.

3. Select **New Organization**.



4. Follow the steps outlined in Section 3.2, "Configuring the Salesforce Connector" to complete the Salesforce connector configuration.

When you have authorized the remote application, you are redirected to Identity Connect. You now have the option to continue with the new organization configuration, or to clone the configuration from an existing organization.

Caution

Creating a new organization configuration by cloning an existing configuration can save time, as you do not have to recreate the mapping rules for profiles and roles. However, you can clone an organization

only if it originates from the same production organization as the original organization. For example, if you have configured a sandbox organization, you can clone this configuration for a new sandbox organization that is based on the same production organization, or for the production organization itself. Within the organization configuration, there are several references to ID values, that are valid only for organizations that are part of the same production organization "family". Attempting to clone an organization configuration across different production organization "families", will cause numerous errors.

Summary Data Source Salesforce Org Reports

Successfully retrieved token from Salesforce!

Would you like to clone an existing organization?

Create as new organization Create Organization

Create as new organization
Clone from Identity Connect

Note

When you clone an organization from an existing organization, the new organization has the identical profile and role mappings that are configured for the existing organization. Permission set and group mappings are not cloned.

By default, live updates and scheduled reconciliation are disabled for a newly cloned organization, regardless of the live update setting for the existing organization. Having updates disabled by default allows you to customize any changes to the new organization configuration before updates start to flow to your Salesforce data. You must manually enable live updates for the cloned organization, as described in Section 5.3, "Configuring the Synchronization Schedule".

- To update the configuration for a particular organization at any stage, return to the Salesforce Org page and select the organization from the dropdown list. When you are configuring the mapping, synchronization, or SSO for your organization, make sure that the organization name displayed in the dropdown list refers to the organization that you are configuring.

3.4. Delete a Salesforce Organization Configuration

To delete the configuration for a particular organization, return to the Salesforce Org > Connection page, select the organization from the dropdown list, and click Delete at the bottom of the page.

If you have already run reconciliation or synchronization operations for this organization, you are asked whether you want to delete the audit data that was generated from these operations.

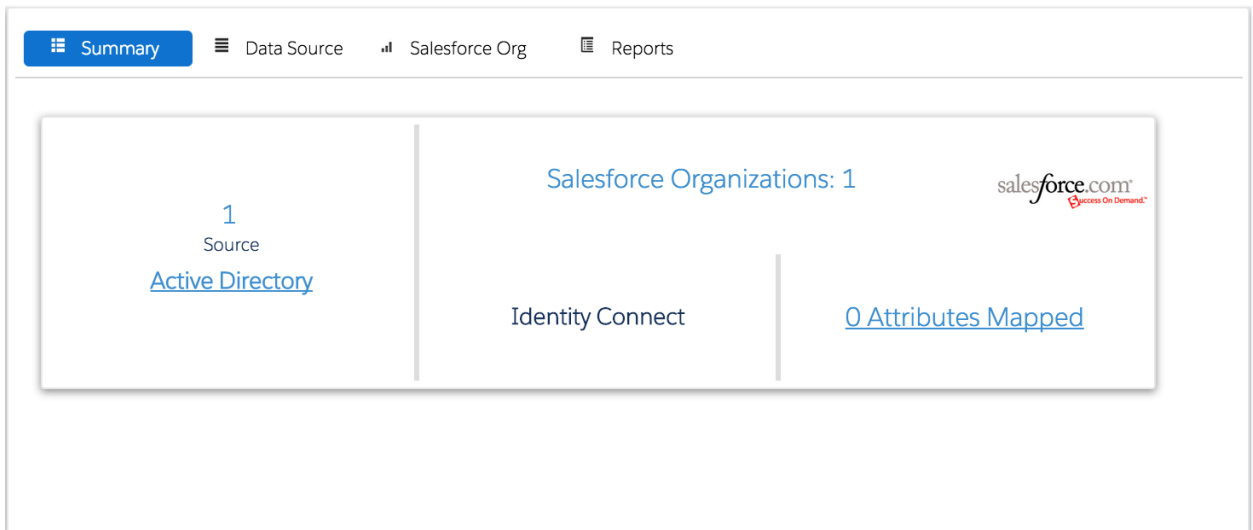
When you delete a Salesforce organization, the following information is removed:

- All user association (link) data corresponding to that organization is deleted from the repository.

- Any data for that organization related to the mapping of permission sets, permission set assignments, permission set license assignments, profiles, groups, and group memberships is deleted from the repository.
- Mappings for that organization are removed from the mapping configuration file ([conf/sync.json](#)).
- The organization is deleted from the organization configuration file ([conf/salesforce.orgs.json](#)).
- Any scheduled tasks for that organization are deleted from the configuration directory, and from the repository.
- If you have elected to delete audit data, any reconciliation audit data relating to that organization is deleted from the repository.

3.5. General Notes About the User Interface

If your UI session times out, you will need to log in again. After you have logged in, the Identity Connect Summary page is displayed. The Summary page gives an indication of where you are in the setup process, and what step must be completed next.



After the Salesforce Connector has been configured, the Summary page displays your Salesforce Organization name and the number of Salesforce organizations that have been configured for this Active Directory instance.

To obtain the exact release version of your Identity Connect instance, click Settings at the top right of the administrative interface and select the About Identity Connect tab.

Customize Theme	SSL Configuration	Authentication and Session	Database	About Identity Connect
-----------------	-------------------	----------------------------	----------	-------------------------------

IC Release Version

Build Revision

IDM Version

Chapter 4

Mapping Data Between Active Directory and Salesforce

Identity Connect enables you to specify how attributes and other data are mapped from the Active Directory data source to the Salesforce data store.

After you have configured the Salesforce connector, click Salesforce Org and then select Mapping on the page for that organization.

The Mapping page covers two main aspects of the mapping of data between Active Directory and Salesforce.

- *Attribute Mapping* maps all the attributes of a user entry to a comparable attribute in Salesforce.

A default set of mapped attributes is presented, with sample values for each attribute. The sample user data is that of the user who is currently logged into Identity Connect. To specify a different sample user from your Active Directory, enter the first few letters of the user name in the Sample User field, click Enter, and select the correct user from the list.

- *Group Mapping* maps Active Directory groups to one or more of the grouping mechanisms within Salesforce.

The following Salesforce grouping mechanisms can be mapped to Active Directory groups:

- *Profiles*. The Profile to AD Group mapping maps Salesforce profiles to Active Directory groups. A default profile mapping is *required*. If a user is not a member of any of the groups that are mapped in this section, he is mapped to the default Salesforce profile.
- *User Roles*. The User Role to AD Group mapping maps Salesforce user roles to Active Directory groups.
- *Permission Sets*. The Permission Set to AD Group mapping maps Salesforce permission sets to Active Directory groups. The permission sets displayed here are those that have been configured for your Salesforce organization.
- *Salesforce Groups*. The SF Group to AD Group mapping maps groups defined within Salesforce to Active Directory Groups.

The first time you open the mapping page, only the Attributes and Profile to AD Group mapping tabs are displayed. You must verify the attribute mapping, and specify a profile to group mapping before you can continue. When you click Save here, the data displayed in both tabs is saved.

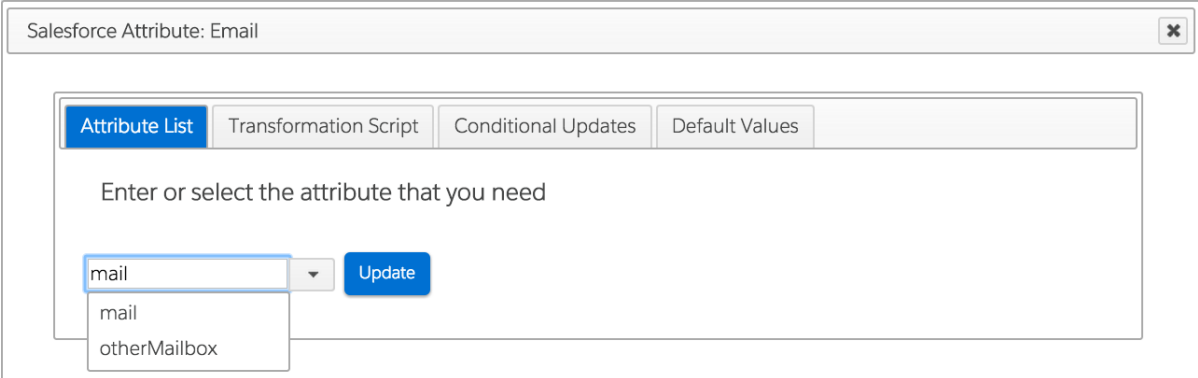
After you have saved the initial two mappings, the remaining Active Directory Group mapping tabs are displayed.

4.1. Mapping Attributes

Attribute mapping enables you to specify how the value of a Salesforce attribute is defined, based on a corresponding Active Directory attribute.

Configure attribute mapping by following these steps:

1. On the Mapping page, select the Attributes tab.
2. Click on the Salesforce attribute whose value you want to define. For example, click on Email to specify the value that will be used for the Salesforce Email attribute.
3. The Attribute List tab enables you to specify an Active Directory attribute to be mapped directly. Enter the name of the attribute or type a few characters of the attribute name to select it from the list.



Salesforce Attribute: Email

Attribute List Transformation Script Conditional Updates Default Values

Enter or select the attribute that you need

mail

mail
otherMailbox

Update

4. The Transformation Script tab enables you to specify how an Active Directory attribute is transformed to provide a value for the Salesforce attribute. The transformation script is a JavaScript that takes a source (Active Directory) attribute, and does something with its value to provide the Salesforce attribute value.

For example, the sample transformation script `source.mail ? source.mail.toLowerCase() : null` takes the value of the `mail` attribute from Active Directory and converts it to lower case to provide the value of the Email attribute in Salesforce. If no value exists for the `mail` attribute, a `null` value is inserted as the value in Salesforce.

The format of the transformation script depends on whether you have selected an attribute on the Attribute List tab. If you do not specify an attribute on the Attribute List tab, the entire object is regarded as the source, and you must include the attribute name in the script (for example, `source.mail.toLowerCase()`). If you specify an attribute on the Attribute List tab, that attribute is regarded as the source, so the transformation script would be `source.toLowerCase()`.

Salesforce Attribute: Email ✕

Attribute List **Transformation Script** Conditional Updates Default Values

Enter a transformation script for complex attributes

Current value for "source":

```
source.mail ? source.mail.toLowerCase() : null
```

Example result:

Use JavaScript code above to perform advanced mapping operations.
This script will be processed for each user that is mapped, to calculate their value for this property.
The result of the final statement in the script will be used as the value for this property.
This script has the "source" variable available to it as input.
The value for the "source" variable depends on how you have defined the attribute mapping in the "Attribute List" tab.
If you haven't selected any value in the "Attribute List" tab, then "source" will be the complete user object with properties for each directory attribute.
If you have selected a value in "Attribute List", then "source" will be that value.

By default, the **manager** attribute in Active Directory is mapped to the **managerID** attribute in Salesforce, using a transformation script. The transformation script locates the Active Directory **manager** property and looks up the manager's **objectGUID**, based on the value of his **distinguishedName**. The script then locates the corresponding SalesforceID of the manager, in the links table, and uses this ID to populate the **managerID** property in Salesforce.

Note

Active Directory attributes can be either single-valued or multi-valued. Multi-valued Active Directory attributes are stored as an array in the connector schema. If you are mapping a Salesforce attribute to a multi-valued Active Directory attribute, your transformation script must take this into account.

For more information about single and multi-valued attributes, see the MSDN article on [Single vs. Multiple Value Attributes](#).

5. For non-mandatory attributes, the Conditional Updates tab enables you to apply two types of conditions that determine specific situations in which an attribute in Salesforce will be updated.
 - You can define a *conditional update script* to prevent an attribute from being updated in Salesforce under certain conditions.

A conditional update script takes the following form:

```
object.attribute operator value
```

where *attribute* refers to the source (Active Directory) attribute. The condition is based on the attribute value of the Active Directory entry. The corresponding attribute in Salesforce is updated *only* if the condition evaluates to true for that entry. The attribute name is case sensitive.

For example, if all users based in Germany worked for a specific department, you might want to prevent any changes to these users' `department` attribute in Salesforce. In this example, you would apply a conditional update script to the `department` attribute in the mapping, which would filter out changes to this attribute for users whose country name (`co`) attribute was `Germany`. The following conditional update script on the `department` attribute would achieve this objective:

```
object.co != "Germany"
```

In other words, update this attribute in Salesforce *only* if the entry's `country` attribute is not `Germany`.

- By default, Salesforce attribute values are set when a user is created, and when that user is updated. You can specify that the attribute value should be set *only* when the user is created, by selecting *Only when creating a new user* from the dropdown list on this tab.

In this case, any updates to a user entry will not reset the Salesforce attribute value for that entry.

Note that, regardless of the situation you select from this list, if you have defined a conditional update script that returns false for an entry, the attribute value will not be set for that entry.

Salesforce Attribute: Country

Attribute List Transformation Script **Conditional Updates** Default Values

Enter a conditional update script for Country

```
object.c != "Germany"
```

Action for user Barbara Jensen: UPDATE

Use JavaScript code above to prevent data from being updated in Salesforce in certain conditions. This script will be processed for each user that is mapped. If the result is true or if there is no script the property will be updated, otherwise updates will not be sent to Salesforce. This script has the "object" variable available to it as a global. The "object" variable represents the complete user record with properties for each source attribute.

If any changes are calculated for this attribute, apply them in these situations:

- When creating and updating a user
- Only when creating a new user

Update

- The Default Values tab enables you to specify a default value that should apply for the Salesforce attribute in the event that the user does not have the corresponding attribute in his Active Directory user entry.

The following example sets a user's `CompanyName` attribute to `example.com` in Salesforce, in the event that the user does not have a value for the `company` attribute in Active Directory.

Salesforce Attribute: CompanyName

Attribute List Transformation Script Conditional Updates **Default Values**

Set a default value for users that may not have the attribute you mapped in their profile

```
example.com
```

Update

7. If the default list of attributes that is presented is not sufficient, click *Add Attribute* to include additional Salesforce attributes in the mapping.

The list of attributes in the Salesforce column is populated directly from the Salesforce data store. You cannot specify your own attributes here, but you can add attributes from the Salesforce list, if the default list does not meet your requirements.

8. To remove an attribute from the mapping, click the Delete icon next to that attribute. Mandatory attributes cannot be removed.
9. When the attribute mapping configuration is complete, click *Save* to save the mapping.

4.2. Mapping Salesforce Profiles to Active Directory Groups

A user's Salesforce profile determines what features that user can access in Salesforce. To specify the Salesforce profile that is applied to an Active Directory user, profiles are mapped to Active Directory groups.

Note

You *must* configure at least one profile to group mapping in order for synchronization to work. It is recommended that you configure a default profile ID value (on the Default Values tab) to ensure that users whose entries do not include the attribute that you mapped are still assigned a profile ID, required for synchronization.

Configure profile mapping by following these steps:

1. On the Mapping page, select the Profile to AD Group tab.
2. On the Attribute List tab, select the LDAP attribute that is used to determine group membership. The default attribute is `memberOf`. Select a different attribute if your organization defines groups in a different way.
3. Select the Value Mapping tab to map profiles directly to Active Directory groups.

The left hand column lists all possible Salesforce profiles. The right hand column indicates the Active Directory groups to which these profiles are mapped. No groups are mapped by default.

Attribute List
Transformation Script
Conditional Updates
Default Values

Enter or select the attribute that you need

Value Mapping
Value Precedence

Salesforce	Active Directory
Chatter Free User	<input type="text" value="None"/>
Chatter External User	<input type="text" value="None"/>
Identity User	<input type="text" value="None"/>
Chatter Only User	<input type="text" value="None"/>
Chatter Moderator User	<input type="text" value="None"/>
Company Communities User	<input type="text" value="None"/>
Customer Community User	<input type="text" value="None"/>
Force.com - App Subscription User	<input type="text" value="None"/>

4. Click the edit icon adjacent to a Salesforce profile to map an Active Directory group to that profile.

You can select more than one Active Directory group to map to the profile. The following selection maps all members of the Active Directory group "salesforceGroups" to the Standard User profile in Salesforce.

memberOf

Value Mapping | Value Precedence

Salesforce	Active Directory
Chatter Free User	<input type="text" value="None"/>
Chatter External User	<input type="text" value="None"/>
Identity User	<input type="text" value="None"/>
Chatter Only User	<input type="text" value="None"/>
Chatter Moderator User	<input type="text" value="None"/>
Company Communities User	<input type="text" value="None"/>
Customer Community User	<input type="text" value="None"/>
Force.com - App Subscription User	<input type="text" value="None"/>
Work.com Only User	<input type="text" value="None"/>
System Administrator	<input type="text" value="None"/>
Read Only	<input type="text" value="None"/>
Standard User	<input type="text" value="salesforceGroups"/> <input type="button" value="Update"/> <input type="button" value="Cancel"/>

- In the event of a user being allocated more than one Salesforce profile, based on the group mapping, you can specify an order of precedence to indicate which profile should be taken into account.

Select the Value Precedence tab to specify the order of precedence. Click and drag the profiles so that they appear in the correct order.

- When you have completed the initial mapping, select the Transformation Script tab to display the transformation that will be applied in order to map the ProfileId. You can also edit this script manually instead of using the preceding tab to generate it.

Note that if you edit the transformation script manually, the Attribute List tab is not updated accordingly. In this case, Identity Connect uses the transformation that is specified on the Transformation Script tab and ignores the Attribute List tab.

7. Select the Default Values tab to specify the default Salesforce profile that will be applied to the user in the event that the user is not a member of any of the Active Directory groups specified here.
8. When you have completed the profile mapping, click Save to apply the mapping.

4.3. Mapping User Roles to Active Directory Groups

A Salesforce Role Hierarchy enables you to define how your organization reports on and accesses data. To map Salesforce roles to groups of Active Directory users, select the User Role to AD Group tab.

User Role Mapping is disabled by default. Select the Enable User Role Mapping checkbox to enable this mapping type.

When User Role Mapping has been enabled, this tab works in a similar way to the Profiles to Groups mapping tab, described in the previous section. The Salesforce roles that are displayed are those that have been defined in your Salesforce organization.

When you have completed the initial mapping, select the Transformation Script tab to display the transformation that will be applied in order to map the user role. You can also edit this script manually instead of using the preceding tab to generate it.

The following selection maps all members of the Active Directory group "Executives" to the Salesforce user role "Executive Staff".

Attributes | Profile to AD Group | User Role to AD Group | Permission Set to AD Group | SF Group to AD Group

Enable User Role Mapping

Attribute List | Transformation Script | Conditional Updates | Default Values

Enter or select the attribute that you need

Value Mapping | Value Precedence

Salesforce	Active Directory
Executive Staff	<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">-</div> <div style="border: 1px solid #ccc; padding: 2px 5px; flex-grow: 1;">Find Groups</div> </div> <div style="border: 1px solid #ccc; padding: 2px 5px; margin-top: 2px;"> Executives </div> <div style="margin-top: 5px;"> Update Cancel </div>
Hardware Sales Rep	<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">✎</div> <div style="border: 1px solid #ccc; flex-grow: 1;">None</div> </div>
Networking Sales Rep	<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">✎</div> <div style="border: 1px solid #ccc; flex-grow: 1;">None</div> </div>
Software Sales Rep	<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">✎</div> <div style="border: 1px solid #ccc; flex-grow: 1;">None</div> </div>
VP Hardware	<div style="display: flex; align-items: center;"> <div style="margin-right: 5px;">✎</div> <div style="border: 1px solid #ccc; flex-grow: 1;">None</div> </div>

4.4. Mapping Permission Sets to Groups

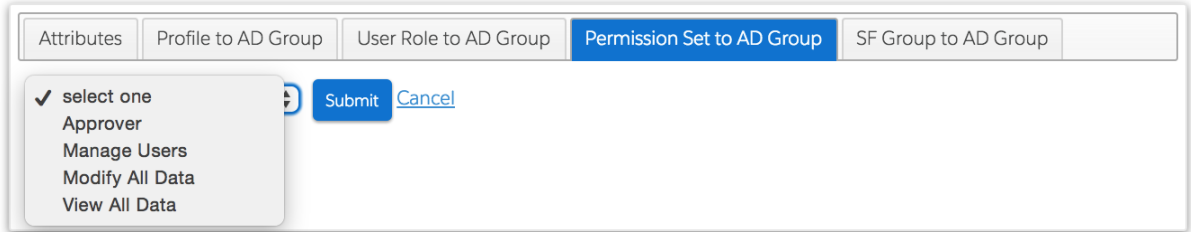
A Salesforce permission set encapsulates a range of settings and permissions that give users access to various areas of a system. You can map Salesforce permission sets to specific groups of users defined in your directory, as follows:

1. On the Mapping page, select the Permission Set to AD Group tab.

Note that this tab is displayed only after you have clicked Save on the "Attributes" or "Profile to AD Groups" tab.

2. No permission sets are displayed by default. To add a permission set to be mapped, click Add Permission Set, select the permission set from the list, and click Submit.

The list of permission sets displayed here are those that you have defined in your Salesforce organization. If you have not created any permission sets in Salesforce (apart from the default Identity Connect permission set), the Add Permission Set button is not displayed.



- When you add new permission sets, the following message is displayed at the bottom of the Permission Set to AD Group page:

Cached Permission Set Assignment Data is OUT OF DATE Update Now

This message indicates that the permission set data that is stored in the Identity Connect repository is out of sync with the new permission set data.

Click Update Now. This action launches a reconciliation operation that synchronizes the permission set data.

Caution

This reconciliation operation will not succeed if all the mapped permission sets are empty. Because the reconciliation operation does not succeed, the mapping continues to display OUT OF DATE, even after the Update Now operation is run. Therefore, before you run the Update Now operation, you *must* ensure that at least one Salesforce user is assigned to one of the selected permission sets.

After the reconciliation operation has completed successfully, a message similar to the following is displayed:

Cached Permission Set Assignment Data Last Updated June 23, 2014 12:57

- Click the edit icon adjacent to the Salesforce permission set to map an Active Directory group to that permission set.

You can select more than one Active Directory group to map to the permission set.

The following image shows how the Manage Users permission set is mapped to the group of Domain Admins in Active Directory.

Attributes Profile to AD Group User Role to AD Group **Permission Set to AD Group** SF Group to AD Group

Add Permission Set

Salesforce Manage Users

Active Directory

Find Groups

● CN=Domain Admins,CN=Users,DC=example,DC=com

Update Cancel

Cached Permission Set Assignment Data Last Updated August 14, 2015 15:10 [Update Now](#)

Click Update to save the new mapping.

- Repeat this procedure for each Salesforce permission set that you want to map.

Note

Mapped permission set assignments are reconciled automatically, every twenty-four hours, at midnight. This regular reconciliation prevents permission set assignments that are made directly in a user's Salesforce account from contradicting the mapping rules that are set up in Identity Connect. If you change a user's permission set assignment directly in Salesforce, that change will eventually (within twenty-four hours) be reverted to match the mapping rules in Identity Connect.

4.5. Permission Set Licenses

Identity Connect tracks the users that it provisions to Salesforce by maintaining a permission set license agreement record for each user. These records correspond with the number of permission set license agreements that your organization has purchased, with one license being "claimed" for each user that is provisioned. If the total number of purchased licenses has been claimed, no additional users can be provisioned.

To obtain the number of available permission set licenses for your Salesforce organization, follow these steps:

- Log in to Salesforce, and select Setup.
- In the Administer section, expand the Company Profile menu and select Company Information.
- Click Permission Set Licenses to display the total licenses purchased, the number of licenses in use (number of users provisioned) and the number of licenses remaining.

Permission Set Licenses					Permission Set Licenses Help ?
Name	Status	Total Licenses	Used Licenses	Remaining Licenses	Expiration Date
Identity Connect	Active	10,000	44	9,956	

If you have reached the maximum number of permission set licenses available, you must either buy additional licenses, or deactivate any defunct users in Salesforce (or disable them in Active Directory) before you can provision any new users.

4.6. Mapping Salesforce Groups to Active Directory Groups

A Salesforce public group is a set of users that can contain individual users, other groups, or the set of users with a specific role.

You can map Salesforce groups to Active Directory groups as follows:

1. On the Mapping page, select the SF Group to AD Group tab.

Note that this tab is displayed only after you have clicked Save on the "Attributes" or "Profile to AD Group" tab.

2. No Salesforce groups are displayed by default. To add a Salesforce group to be mapped, click Add SF Group, select the Salesforce group from the list, and click Submit.

The list of groups displayed here are those that have been defined in your Salesforce organization.

3. When you add new Salesforce groups, the following message is displayed at the bottom of the SF Group to AD Group page:

Cached SF Group Membership Data is OUT OF DATE Update Now

This message indicates that the Salesforce Group data that is stored in the Identity Connect repository is out of sync with the new Salesforce Group data.

Click Update Now. This action launches a reconciliation operation that synchronizes the Salesforce Group data.

Caution

This reconciliation operation will not succeed if all the mapped Salesforce Groups are empty. Because the reconciliation operation does not succeed, the mapping continues to display OUT OF DATE, even after the Update Now operation is run. Therefore, before you run the Update Now operation, you *must* add at least one user to at least one mapped Salesforce Group.

After the reconciliation operation has completed successfully, a message similar to the following is displayed:

Cached SF Group Membership Data Last Updated June 23, 2014 13:24

4. Click the edit icon adjacent to the Salesforce group to map an Active Directory group to that Salesforce group.

You can select more than one Active Directory group to map to the Salesforce group.

The following image shows how the example-employees Salesforce group is mapped to the group of Employees in Active Directory.

Attributes Profile to AD Group User Role to AD Group Permission Set to AD Group **SF Group to AD Group**

Add SF Group

Salesforce	Active Directory
example-employees	CN=Employees,DC=example,DC=com

Cached SF Group Membership Data Last Updated August 17, 2015 14:26 [Update Now](#)

Clear Changes Save

Click Update to save the new mapping.

5. Repeat this procedure for each Salesforce group that you want to map.

Chapter 5

Data Synchronization and User Association Management

The main purpose of Identity Connect is to maintain data consistency between your Active Directory and your Salesforce data store. This consistency is achieved by a process called *synchronization*, which modifies user data on the target system (Salesforce) to match the data in Active Directory.

This section provides an overview of the synchronization process and walks you through the synchronization configuration to establish associations between user entries.

5.1. Overview of the Synchronization Process

Synchronization changes user data on a target system so that it matches the data on a source system. Before synchronization can occur, a *reconciliation* report is run. Reconciliation is the process by which two data sources are assessed and the consistency of the data across the two systems is analyzed. Part of the reconciliation process involves identifying the user accounts that exist in the two data stores, and assessing their potential for matching.

After a reconciliation run, the Reconciliation Report identifies all user and group accounts and categorizes them, based on the extent to which a match is found between the source and the target. User accounts are divided into two main categories:

- *Valid Active Directory Users* are user accounts that exist in Active Directory and are candidates for synchronization. A valid AD user account can be one of the following:
 - *1-1 Match*, meaning that a clear and unique match exists in Salesforce, with no ambiguity.
 - *No Match Found*, meaning that there is no corresponding entry in Salesforce, although the Active Directory user is a valid user for synchronization.

In an initial provisioning process (before the Salesforce organization has been populated with entries), this is the most likely situation for AD user entries. Entries are unlikely to be in this state if a scheduled synchronization or liveSync process has been configured.

- *Conflicting Match*, meaning that more than one potential match exists in Salesforce. Entries in this category should be manually assigned to the correct Salesforce user.
- *Other Users* are all entries in either Active Directory or Salesforce that are not candidates for synchronization. This category normally indicates "orphan entries" in either the source or target data stores. Other users can be one of the following:

- *Unresolved AD Users* are user accounts that exist in Active Directory but either have no match in Salesforce or the potential Salesforce match has already been associated with another Active Directory user account.

Most commonly, entries fall into this category when an Active Directory entry that was previously linked to a Salesforce entry, has lost its corresponding entry, but the link was not removed. Alternatively, if a manual user assignment has already been made to the corresponding Salesforce entry, this link can prevent the correct Active Directory entry from being mapped.

- *Unresolved SF Users*, are user accounts that exist only in Salesforce and not in Active Directory.

Generally, the corresponding Active Directory entry is missing, either because it never existed, or because it was removed and the change has not yet been picked up by synchronization or liveSync.

- *Ignored Users*, are user accounts for which no match exists but which are not cause for concern.

The existence of these users in only one data source (either Active Directory or Salesforce) is expected, and the accounts are ignored in future synchronization reports and reconciliation runs.

For example, certain Salesforce administrative entries might be required only in the context of Salesforce administration and have no use in Active Directory. These *known unresolved entries* can be flagged so that they appear in a separate list in future synchronization runs. Separating the ignored entries from the unresolved entries ensures that the list of unresolved entries remains a priority for cleanup.

Ignoring a user that has previously been synchronized with Salesforce releases the user's Identity Connect license in Salesforce. However, "unignoring" the user does not reestablish that license. A new synchronization operation is required to reestablish the user's Identity Connect license.

In general, unresolved Active Directory and Salesforce entries are cleaned up during synchronization. Entries that exist only in Active Directory are created in Salesforce. Entries that exist only in Salesforce are deactivated (in the event that they have been deleted from Active Directory) or must be moved manually to the list of Ignored Users (if they have never existed in Active Directory and are not candidates for synchronization).

Inactive Salesforce users are filtered out of the reconciliation process. However, they are still visible in Identity Connect and can therefore be manually linked to Active Directory entries and reactivated.

5.2. Managing User Associations

When a reconciliation operation finds a matching target entry, a *link* is created between the source and the target entry. This link is referred to in Identity Connect as a *User Association*. User associations serve two purposes - they speed up future reconciliation operations, and they serve as a record of a source or target entry's existence.

For example, a target entry might be deleted at some point, but if an association to the source entry still exists, there is evidence that the target entry once existed. This functionality is useful for auditing purposes. If there is conflicting data between two matched entries, the reconciliation operation might be unable to associate the entries. In this case, the entries can be associated manually.

The first time you configure synchronization, Identity Connect performs a blank reconciliation run. In this initial operation, no records are changed in Salesforce. The reconciliation report that is generated enables you to assess the consistency of the entries stored in Active Directory with those stored in Salesforce, by automatically associating user entries, wherever possible. Based on this report, you can change the Default User Association Rules, where necessary. Note that the Default Association Rules only apply before the first real synchronization operation is performed (that is, before any links exist between entries in Active Directory and entries in Salesforce). You should review the associations carefully before running a real synchronization operation for the first time because it is more difficult to isolate or fix inaccurate associations after the data has been synchronized.

Entries that could still not be associated automatically, after the association rules have been finalized, can be associated manually, prior to performing a real synchronization operation. In the ideal scenario, all entries are either associated, or have been marked as "Ignored".

5.2.1. Changing User Associations Manually

Clean Data refers to all entries in the source and target system being matched and associated, with no conflicts in the entry data, and known unmatched entries being marked as ignored. The reconciliation report indicates the percentage of data that is *clean*. For more information, see Section 9.1, "Running Reconciliation Reports".

Identity Connect provides a mechanism to clean up data by working through any unmatched, unassociated, or conflicting entries found during the reconciliation run. The following list describes the data cleanup process for each category into which user entries fall after a reconciliation run.

1-to-1 Match

Generally, valid AD Users with a 1-to-1 Match do not require manual intervention. However, there might be specific entries whose user associations have been made incorrectly by the automatic association mechanism. In this case, you can manually disassociate these user entries and reassociate them to the correct entry.

To change a user association manually, see Procedure 5.1, "To Create or Change User Associations Manually".

No Match Found

For valid AD Users for whom no Salesforce match was found, you can use manual association to link the entry to an existing Salesforce account or move the entry to the list of ignored entries. Ignored entries will not appear in the list of unmatched entries in any future reconciliation reports. If you do not make any manual association on an unmatched account, the account is created in Salesforce when the data is synchronized.

To locate a match in Salesforce manually, see Procedure 5.1, "To Create or Change User Associations Manually". The manual association will be used during future synchronization runs. To move the entry to the list of Ignored Users, see Procedure 5.2, "To Move Users to the Ignored List".

Conflicting Matches

Conflicting matches are user entries for which more than one potential match exists. To resolve conflicting matches, specify a match for the user manually, as described in Procedure 5.1, "To Create or Change User Associations Manually".

Unresolved AD Users

These users exist only in Active Directory, with no known match in Salesforce. You can either find a match in Salesforce manually (see Procedure 5.1, "To Create or Change User Associations Manually") or move the entry to the list of Ignored Users (see Procedure 5.2, "To Move Users to the Ignored List").

Unresolved SF Users

These users exist only in Salesforce, with no known match in Active Directory. You can either find a match in Active Directory manually (see Procedure 5.1, "To Create or Change User Associations Manually") or move the entry to the list of Ignored Users (see Procedure 5.2, "To Move Users to the Ignored List").

Ignored Users

Only users that you have explicitly moved to the ignored list appear here. If you have moved a user to the ignored list in error, select that user and click Change User Association to move the user out of the ignored list and manually associate it with its matching entry (see Procedure 5.1, "To Create or Change User Associations Manually").

Procedure 5.1. To Create or Change User Associations Manually

To create a user association manually, or to change an association that was created automatically, follow these steps:

1. On the Sync page, select the tab and category for the user entry that you want to associate.

Select the entry that you want to match, or whose association you want to change, and click Change User Association.
2. In the Change User Association window, select an item by which to search for the correct user in either Salesforce or Active Directory. You can search by Alias, Email, First Name or Last Name.

Enter the required value for this field and click Search.
3. All entries that match your search are displayed underneath the Search button. Select the correct entry to be matched, and click Link Account.

4. The user is now associated with the account that you have selected, rather than with the account that was identified during automatic association.

This manual association will override the automatic association during future synchronization runs.

Procedure 5.2. To Move Users to the Ignored List

To move a user to the list of ignored users, follow these steps:

1. On the Sync page, select the tab and category for the user entry or entries that you want to ignore.
2. Select the entry or entries and click Ignore User(s).

Users that have been moved to this category are displayed in the list of Ignored Users on the Other Users tab.

Procedure 5.3. To Test the Mapping for a Specific Entry

After you have completed the mapping configuration and established an association between two entries, either automatically or manually, test the configuration for a specific user as follows:

1. On the Sync page, select the tab and category for the user entry that you want to test.
2. Select the entry and click Sync Selected Record.

The Single User Data Synchronization panel displays the record in Active Directory and the current corresponding record in Salesforce, if there is one.

Click Sync Now to perform the synchronization operation on that particular record.

If the synchronization is unsuccessful, an error indicating the reason for the failure is displayed at the top of the screen. You can use this information to correct any errors in the mapping, and attempt the synchronization test again.

Note that this step synchronizes the data of this particular user - it is not merely a validity check. Synchronizing a single user enables you to test your mapping before applying it to the entire Salesforce data store.

The following image shows an Active Directory new user entry that did not exist in Salesforce until the Sync Now operation was launched.

Single User Data Synchronization
✕

Active Directory Properties

Current values

adminDisplayName	<input type="text"/>
c	<input type="text"/>
cn	<input type="text" value="Stephen Carter"/>
co	<input type="text"/>
company	<input type="text"/>

Salesforce Properties

Before sync		After sync
FirstName	<input type="text" value="User not found"/>	<input type="text" value="Stephen"/>
LastName	<input type="text" value="User not found"/>	<input type="text" value="Carter"/>
Email	<input type="text" value="User not found"/>	<input type="text" value="scarter@example.com"/>
Username	<input type="text" value="User not found"/>	<input type="text" value="scarter@example.com"/>
Alias	<input type="text" value="User not found"/>	<input type="text" value="scarter"/>

[Sync Now](#)

After you have changed the automatic user associations and moved any entries to the Ignored Users list, click Analyze Associations Now to run the reconciliation report again. Verify that all the user associations are correct before enabling synchronization for the entire data store.

To synchronize data immediately, after you have verified the user associations, click Sync Now.

Depending on the number of records in your directory, a complete synchronization operation can take some time. A status bar indicates the progress of the synchronization operation.

Scheduled Data Synchronization

Schedule Updates - Schedule a recurring process to sync all of your user data, for full consistency.

Every at minutes past the hour

Live Updates - Information will be synced with Salesforce as soon as it is changed in your directory.

⌂ Sync Now

37%

Save

5.2.2. Association Rules

Association Rules are the criteria by which user accounts are mapped between Salesforce and Active Directory. By default, users are linked if one of the following situations is true:

- Their email addresses match
- Their first name and last name match and either their phone, mobile phone, or title match.

You can change the way in which these matches are made by changing the Association Rules on the Sync Page.

Click Change Association Rules and add or remove fields to compile your own set of association rules.

5.3. Configuring the Synchronization Schedule

Data synchronization enables you to specify when and how often Active Directory data changes are pushed to the Salesforce data store. Data can be synchronized according to a defined schedule, or automatically, as soon as changes are made in Active Directory.

The Live Updates mechanism is intended to react quickly to changes as they happen. Live Updates are, however, a best effort mechanism that can miss changes in certain situations. In addition, if a system is down when an update occurs on Active Directory, that change might not be propagated to Salesforce when the system comes back online.

When you use the Live Updates mechanism, you must configure Identity Connect against a single domain controller. Alternatively, for high availability, configure Identity Connect through a load balancer that is sticky to a specific domain controller. Because Live Updates rely on the `uSNChanged` timestamp (which is not replicated), if you do not configure Identity Connect against a single Domain Controller, updates will be missed.

Scheduled Updates are more thorough. The Scheduled Updates mechanism recognizes system error conditions and catches changes that might be missed by the Live Updates mechanism.

Scheduled Updates consume at least one Salesforce API limit per user per scheduled run. Live Updates consume API limits only when a change is pushed to Salesforce. Therefore, you should enable both Scheduled Updates and Live Updates in production. Use Live Updates as the primary syncing mechanism, and Scheduled Updates no more than once per day to ensure appropriate change coverage while limiting the API usage against your org.

Note

When you are working with multiple Salesforce organizations, the live update (or LiveSync) configuration and the scheduled update (reconciliation) configurations apply only to the Salesforce organization for which they are configured. That is, you must explicitly configure live updates and scheduled updates for each separate organization. This is the case even if you have cloned an organization configuration from an existing organization.

To configure the synchronization schedule, follow these steps:

1. Select the Sync tab for the Salesforce organization that you are configuring.
2. In the Scheduled Data Synchronization area, check *Schedule Updates* to specify a regular schedule for synchronization.
3. Select an update interval from the drop down list. Selecting `minute`, `hour`, `day`, and so on, specifies that updates are scheduled once every minute, hour or day. Selecting `(n) days`, `(n) hours`, and so on, enables you to specify the precise number of days, hours or minutes between each update.
4. Select *Live Updates* to indicate that data should be synchronized as soon as changes are made in Active Directory.
5. Click *Save* to save the synchronization configuration.

5.4. Increasing the Number of Connections for Multiple Synchronizations

In high latency environments, running several concurrent synchronization operations can sometimes cause connection errors between Identity Connect and Salesforce. In this situation, you might see an error similar to the following:

```
Unable to find a connection to send the request
```

This error can generally be resolved by increasing the number of connections available to Identity Connect. Increase the number of connections as follows:

1. Change to the Identity Connect configuration directory.

```
$ cd /path/to/salesforceIdConnect/conf
```

2. Using a text editor, edit the configuration file for your Salesforce organization (`salesforce-org-id.json`), adding the following properties:

- `"maxConnectionsPerHost"` - the maximum number of simultaneous requests that a single Identity Connect browser session can make to each Salesforce site (Token and API)
- `"maxTotalConnections"` - the total maximum number of connections maintained in the connection pool

If these properties are not set, their default values are `10` and `20` respectively. Increase the values of these parameters to make more connections available to Identity Connect. Generally, if you are seeing connection errors, doubling the value of both parameters will prevent such errors.

The following excerpt of an edited organization configuration file shows the parameters set to `20` and `40` respectively.

```
$ more salesforce-00DS0000003K4fU.json
{
  "enabled" : true,
  "configurationProperties" : {
    ...
    "idleCheckInterval" : 10000,
    "connectTimeout" : 120000,
    "maxConnectionsPerHost" : 20,
    "maxTotalConnections" : 40,
    "instanceUrl" : "https://example-com.my.salesforce.com",
    ....
  }
}
```

Chapter 6

Configuring Single Sign-On

Identity Connect enables you to set up single sign-on (SSO) using the Security Assertion Markup Language (SAML). With SSO configured, when a user accesses his Salesforce organization URL, he is redirected to the Identity Connect user interface (at <https://hostname.domain.com:8443/connect/>). Logging in to this interface routes the user directly to his Salesforce dashboard.

Salesforce does not validate the user's password. Identity Connect validates the user's credentials (with a simple username/password check or by using Kerberos) and generates an assertion that is sent back to Salesforce in an HTTP POST request. Salesforce then verifies the assertion and allows single sign-on if the assertion is true. The assertion includes an identifier (in the `NameIdentifier` element of the `Subject` statement) which maps to the Federation ID attribute value. This value is populated during a synchronization operation.

You can enable SSO automatically (allowing Identity Connect to complete the SAML configuration), or manually (completing the SAML configuration in Salesforce yourself).

Before you start, make sure that you have registered a domain in Salesforce, as described in Section 2.5.2, "Registering a Domain in Salesforce".

Procedure 6.1. To Enable SSO Automatically

The easiest way to configure SSO is to allow Identity Connect to create the SAML configuration for you. If SAML has been enabled for your Salesforce organization, Identity Connect automatically generates the required SSO configuration the first time it connects to your Salesforce organization. Even if there is no SAML configuration entry in your Salesforce organization, Identity Connect populates it for you during configuration.

1. To create the SAML configuration, select the Salesforce Org tab in the Identity Connect administrative interface, select the organization for which you are configuring SSO, and click SSO.

Identity Connect ▾

Connection

Mapping

Sync

SAML Configuration

Automatic SSO configuration available

	Current SAML Configuration in Salesforce	Expected SAML Configuration for Identity Connect
Name	<input type="text"/>	
Entity Id	<input type="text"/>	https://saml.salesforce.com
Identity Provider Login URL	<input type="text"/>	https://identityconnect.example.com
Identity Provider Logout URL	<input type="text"/>	https://identityconnect.example.com
Issuer	<input type="text"/>	https://identityconnect.example.com
SAML Identity Location	<input type="text"/>	SubjectNameId
SAML Identity Type	<input type="text"/>	FederationId
SAML Version	<input type="text"/>	SAML2_0
Service Provider Initiated Request Binding	<input type="text"/>	true
User Provisioning Enabled	<input type="text"/>	false

Create SAML Configuration Now

The SSO page displays the current SAML configuration in Salesforce, and the configuration that is expected by Identity Connect.

2. Click Create SAML Configuration Now.
3. Identity Connect updates the single sign-on settings in your Salesforce organization. The updated config is displayed in Identity Connect.

SAML Configuration

Automatic SSO configuration available

	Current SAML Configuration in Salesforce	Expected SAML Configuration for Identity Connect
Name	Identity Connect	Hide read-only properties
Entity Id	https://saml.salesforce.com	https://saml.salesforce.com
Identity Provider Login URL	https://identityconnect.example.com	https://identityconnect.example.com
Identity Provider Logout URL	https://identityconnect.example.com	https://identityconnect.example.com
Issuer	https://identityconnect.example.com	https://identityconnect.example.com
SAML Identity Location	SubjectNameId	SubjectNameId
SAML Identity Type	FederationId	FederationId
SAML Version	SAML2_0	SAML2_0
Service Provider Initiated Request Binding	true	true
User Provisioning Enabled	false	false

Update SAML Configuration Now
SAML Config Last Modified: August 17, 2015 14:33

Regular users should now be routed to their Salesforce dashboard when they access the Identity Connect user interface. Note that you must have run at least one synchronization operation for the Federation ID attribute value to be populated for each Active Directory user.

Chapter 7

Configuring Identity Connect for Integrated Windows Authentication (Advanced Feature)

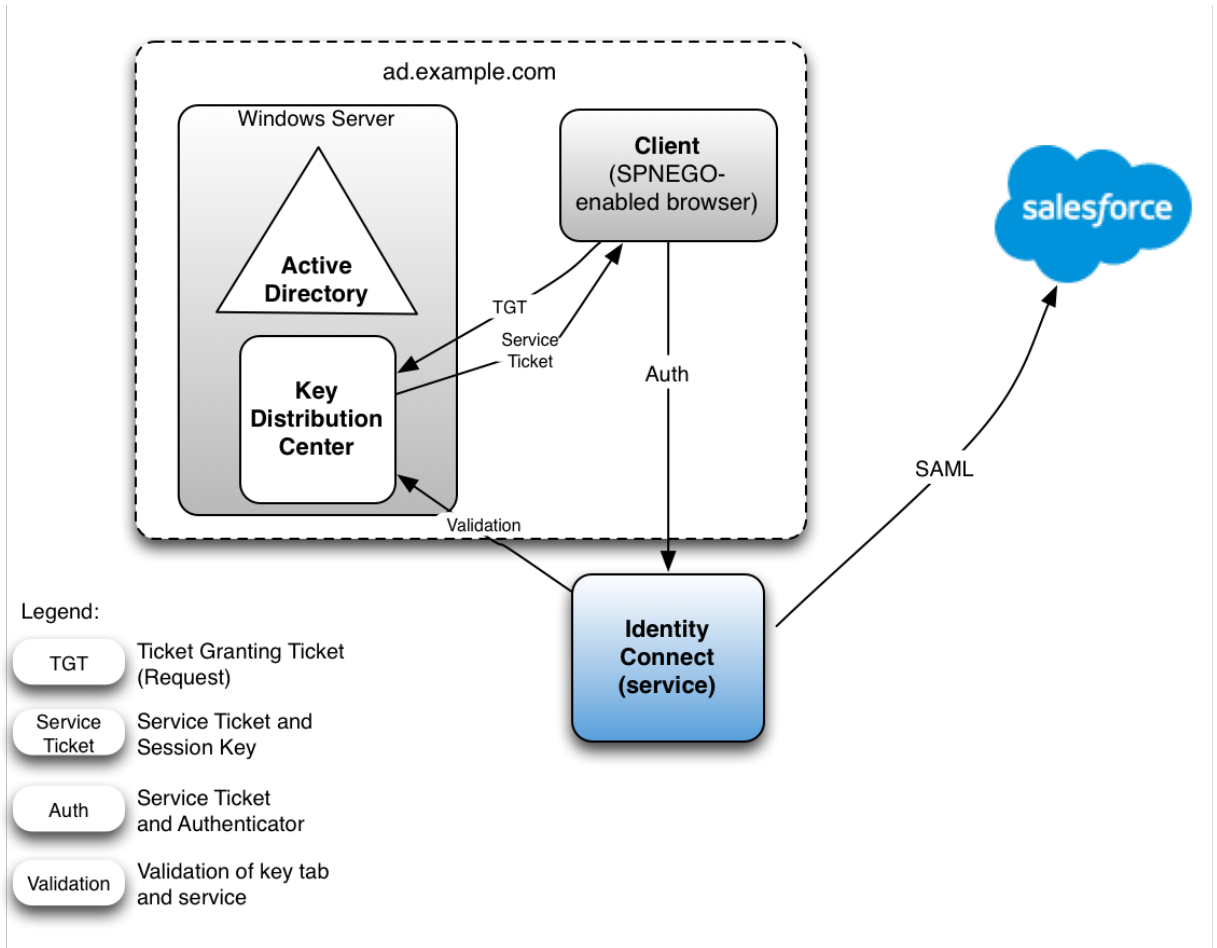
Caution

This feature requires advanced knowledge of Active Directory and Kerberos-based Authentication. Please ensure you have these expertise internally or engage with an implementation partner when implementing this feature.

You can configure Identity Connect so that clients use Integrated Windows Authentication (IWA), rather than authenticating by providing a username and password.

This chapter describes the steps required to use IWA with Identity Connect. The chapter assumes that you are familiar with the principles of IWA, Kerberos and SPNEGO.

The following figure outlines the components involved in configuring Identity Connect for IWA. This setup assumes that the Active Directory server, Identity Connect, and the clients (SPNEGO-enabled browsers) run on different hosts, as indicated in the following diagram. If they run on the same host, the Kerberos ticket cannot be issued to the clients.



This example assumes that the Kerberos Key Distribution Center and the clients that are requesting tickets are in the same Windows domain. This might not always be the case. The examples at the end of this section illustrate additional scenarios, when the client is not part of the domain.

The following sections describe the process for configuring Identity Connect to use IWA. The process includes three broad steps:

1. Configure a Kerberos user account and create a keytab file

Perform this step on the machine that hosts the KDC (your Active Directory server host).

2. Configure the authentication filter in Identity Connect

This step changes the Identity Connect configuration. View the Identity Connect UI in any browser to perform this step.

3. Configure the client browser to support SPNEGO

Perform this step on every client machine that will access Salesforce via Identity Connect at <https://hostname.domain:8443/connect/>.

If you encounter problems during this process, see Section 14.1, "Troubleshooting the Integrated Windows Authentication Configuration".

7.1. Before You Start

Before you start setting up IWA with Identity Connect, make sure that the following steps are in place:

- Identity Connect is installed, configured, and working correctly.
- The basic SSO configuration (without IWA) is working correctly. Do not try to set up IWA if basic SSO is not working - first resolve those issues. For more information, see Chapter 6, "*Configuring Single Sign-On*".

7.2. Configuring the Kerberos User and Creating the Keytab

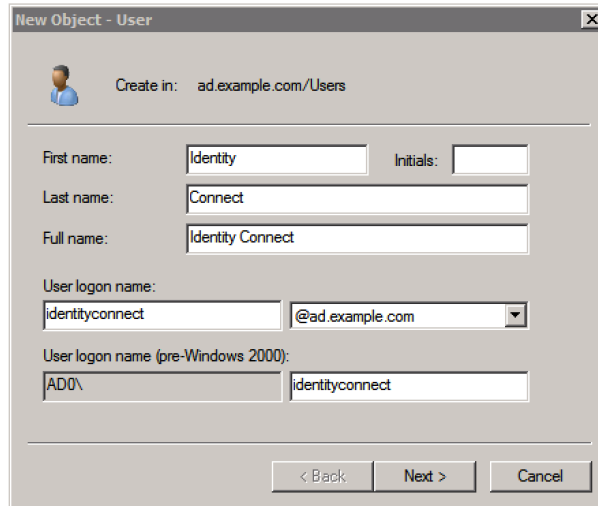
7.2.1. Creating a Specific Kerberos User Account for Identity Connect

To authenticate Identity Connect to the Kerberos Key Distribution Centre (KDC) you must create a specific user entry in Active Directory whose credentials will be used for this authentication. This Kerberos user account should not be used for anything else, that is, it must be a separate user account to the one that Identity Connect uses to connect to Active Directory.

The Kerberos user account is used to generate the Kerberos keytab. If you change the password of this Kerberos user after you have set up IWA, you must update the keytab accordingly.

Create a new user in Active Directory as follows:

1. Provide a login name for the user that reflects its purpose, for example, `identityconnect@ad.example.com`.



New Object - User

Create in: ad.example.com/Users

First name: Identity Initials:

Last name: Connect

Full name: Identity Connect

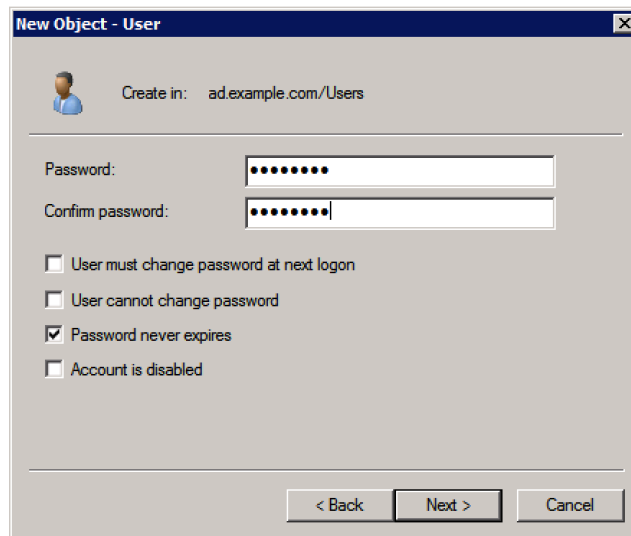
User logon name: identityconnect @ad.example.com

User logon name (pre-Windows 2000): AD0\ identityconnect

< Back Next > Cancel

2. Enter a password for the user. Check the *Password never expires* option and leave all other options unchecked.

If the password of this user account expires, and is reset, you must update the keytab with the new password. It is therefore easier to create an account with a password that does not expire.



New Object - User

Create in: ad.example.com/Users

Password:

Confirm password:

User must change password at next logon

User cannot change password

Password never expires

Account is disabled

< Back Next > Cancel

3. Click Finish to create the user.

7.2.2. Creating the Keytab File

A Kerberos keytab file (`krb5.keytab`) enables Identity Connect to validate the Kerberos tickets that it receives from client browsers. You must create a Kerberos keytab file for the host on which Identity Connect is running.

This section describes how to use the `ktpass` command, included in the Windows Server toolkit, to create the keytab file. Run the `ktpass` command on the Active Directory domain controller. Pay close attention to the use of capitalization in this example because the keytab file is case-sensitive.

The following command creates a keytab file (named `identityConnect.HTTP.keytab`) for the Identity Connect service located at `connect.ad.example.com`.

```
C:\Users\Administrator>ktpass ^
-princ HTTP/connect.ad.example.com@AD.EXAMPLE.COM ^
-mapUser AD\identityconnect ^
-mapOp set ^
-pass Passw0rd1 ^
-crypto ALL
-pType KRB5_NT_PRINCIPAL ^
-kvno 0 ^
-out identityConnect.HTTP.keytab

Targeting domain controller: host.ad.example.com
Using legacy password setting method
Successfully mapped HTTP/connect.ad.example.com to identityconnect.
Key created.
Output keytab to identityConnect.HTTP.keytab:
Keytab version: 0x502
keysize 79 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x1 (DES-CBC-CRC) keylength 8 (0x73a28fd307ad4f83)
keysize 79 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x3 (DES-CBC-MD5) keylength 8 (0x73a28fd307ad4f83)
keysize 87 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x17 (RC4-HMAC) keylength 16 (0xa87f3a337d73085c45f9416b
e5787d86)
keysize 103 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_N
T_PRINCIPAL) vno 0 etype 0x12 (AES256-SHA1) keylength 32 (0x6df9c282abe3be787553
f23a3d1fcef6c6fc4a29c3165a38bae36a8493e866d60)
keysize 87 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x11 (AES128-SHA1) keylength 16 (0xf616977f071542cd8ef3f
f4e2ebcc09c)
```

The `ktpass` command takes the following options:

- `-princ` specifies the service principal name in the format `service/host-name@realm`

In this example (`HTTP/connect.ad.example.com@AD.EXAMPLE.COM`), the client browser constructs an SPN based on the following:

- The service name (HTTP).

The service name for SPNEGO web authentication *must* be HTTP.

- The FQDN of the host on which Identity Connect runs (`connect.ad.example.com`).

This example assumes that users will access Identity Connect at the URL <https://connect.ad.example.com:8443/connect/>.

- The Kerberos realm name ([AD.EXAMPLE.COM](#)).

The realm name must be in upper case. A Kerberos realm defines the area of authority of the Kerberos authentication server.

- `-mapUser` specifies the name of the Kerberos user account to which the principal should be mapped (the account that you created in Section 7.2, "Configuring the Kerberos User and Creating the Keytab"). In our example, the Kerberos user name is `identityconnect`.
- `-mapOp` specifies how the Kerberos user account is linked. Use `set` to set the first user name to be linked. The default (`add`) adds the value of the specified local user name if a value already exists.
- `-pass` specifies a password for the principal user name. Use `"*"` to prompt for a password.
- `-crypto` Specifies the cryptographic type of the keys that are generated in the keytab file. Use `ALL` to specify all crypto types.

This procedure assumes a 128-bit cryptosystem, with a default RC4-HMAC-NT cryptography algorithm. You can use the `ktpass` command to view the crypto algorithm, as follows:

```
C:\Users\Administrator> ktpass -in .\identityConnect.HTTP.keytab
Existing keytab:
Keytab version: 0x502
keysize 79 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 0 etype 0x1 (DES-CBC-CRC) keylength 8 (0x73a28fd307ad4f83)
keysize 79 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 0 etype 0x3 (DES-CBC-MD5) keylength 8 (0x73a28fd307ad4f83)
keysize 87 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 0 etype 0x17 (RC4-HMAC) keylength 16 (0xa87f3a337d73085c45f9416be5787d86)
keysize 103 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 0 etype 0x12 (AES256-SHA1) keylength 32 (0x6df9c282abe3be787553f23a3d1fcef6
fc4a29c3165a38bae36a8493e866d60)
keysize 87 HTTP/connect.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT_PRINCIPAL)
vno 0 etype 0x11 (AES128-SHA1) keylength 16 (0xf616977f071542cd8ef3ff4e2ebcc09c)
```

If your company's Active Directory server requires a higher strength cryptosystem, such as [AES256-SHA1](#), see Use of a high strength cipher for the keytab.

- `-ptype` Specifies the principal type. Use `KRB5_NT_PRINCIPAL`.
- `-kvno` specifies the key version number. Set the key version number to 0.
- `-out` specifies the name of the keytab file that will be generated. Use `identityConnect.HTTP.keytab`.

Note that the keys that are stored in the keytab file are similar to user passwords. You must therefore protect the Kerberos keytab file in the same way that you would protect a file containing passwords.

For more information about the `ktpass` command, see the [ktpass](#) reference in the Windows server documentation.

7.2.3. Setting Up The Keytab File For a Load Balanced Deployment

If you are using Identity Connect behind a load balancer, you must create an additional Kerberos user account for the load balancer, and a keytab file for the load balancer host. Create a new user account, as described in Section 7.2.1, "Creating a Specific Kerberos User Account for Identity Connect". The examples in this section assume that the load balancer account is `lb-user@ad.example.com`.

Map this new account to an SPN, then create a keytab file for the load balancer SPN.

The following command maps the account (`lb-user`) to an SPN (`HTTP/lb.ad.example.com`) and creates a keytab file (`lb.HTTP.keytab`) for the load balancer located at `lb.ad.example.com`.

```
C:\Users\Administrator>ktpass ^
-princ HTTP/lb.ad.example.com@AD.EXAMPLE.COM ^
-mapUser AD\lb-user ^
-mapOp set ^
-pass Passw0rd1 ^
-crypto ALL
-pType KRB5_NT_PRINCIPAL ^
-kvno 0 ^
-out lb.HTTP.keytab

Targeting domain controller: host.ad.example.com
Using legacy password setting method
Successfully mapped HTTP/lb.ad.example.com to lb-user.
Key created.
Output keytab to lb.HTTP.keytab:
Keytab version: 0x502
keysize 79 HTTP/lb.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x1 (DES-CBC-CRC) keylength 8 (0x73a28fd307ad4f83)
keysize 79 HTTP/lb.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x3 (DES-CBC-MD5) keylength 8 (0x73a28fd307ad4f83)
keysize 87 HTTP/lb.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x17 (RC4-HMAC) keylength 16 (0xa87f3a337d73085c45f9416b
e5787d86)
keysize 103 HTTP/lb.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_N
T_PRINCIPAL) vno 0 etype 0x12 (AES256-SHA1) keylength 32 (0x6df9c282abe3be787553
f23a3d1fcefc6fc4a29c3165a38bae36a8493e866d60)
keysize 87 HTTP/lb.ad.example.com@AD.EXAMPLE.COM ptype 1 (KRB5_NT
_PRINCIPAL) vno 0 etype 0x11 (AES128-SHA1) keylength 16 (0xf616977f071542cd8ef3f
f4e2ebcc09c)
```

7.3. Configuring the Authentication Filter in Identity Connect

IWA is disabled in Identity Connect by default. Follow this procedure to enable IWA and to configure the authentication filter.

1. Log in to the Identity Connect administrative interface (for example <https://connect.ad.example.com:8443/admin>).
2. Click Settings in the top right corner and select the Authentication and Session tab.
3. Select Enable Integrated Windows Authentication.

4. Enter the following information:

- **Kerberos Distribution Center.** Enter the FQDN or IP address of the Key Distribution Center (KDC), for example, `host.ad.example.com`.

For a single domain, this is usually the host machine on which your Active Directory server is installed. For multiple domains, check with your IT administrator for the value of this field.

- **Kerberos Realm.** Enter the name of the Kerberos realm, in upper case, for example, `AD.EXAMPLE.COM`.

Note that this is the Kerberos realm described in Section 7.2.2, "Creating the Keytab File".

- **Service Principal Name (SPN).** Enter the SPN of the Kerberos user account that you created previously. This value *must* match the output that was obtained during the keytab creation. In our example, this would be `HTTP/connect.ad.example.com@AD.EXAMPLE.COM`.

If your service is behind a load balancer, enter the SPN of the load balancer user account here. In the previous example, this would be `HTTP/lb.ad.example.com@AD.EXAMPLE.COM`.

- **Upload Kerberos Keytab File.** Click Browse to locate the keytab file that you created in the previous section.

Customize Theme | SSL Configuration | **Authentication and Session** | Database | About Identity Connect

Enable Integrated Windows Authentication?

IWA Details

Kerberos Distribution Center
host.ad.example.com

Kerberos Realm
AD.EXAMPLE.COM

Service Principal Name (SPN)
HTTP/connect.ad.example.com@AD.EXAMPLE.COM

Upload Kerberos Keytab File
Choose File | identityConnect.HTTP.keytab.rtf

If your service is behind a load balancer, browse for the keytab file that you created for the load balancer host.

In a clustered environment, you must copy this load balancer keytab file to the `/path/to/salesforceIdConnect/security` folder on all additional Identity Connect nodes.

Note

Name resolution must be valid for the server that is specified in the **Kerberos Distribution Center** field. If this is not the case, Identity Connect will be unable to contact the KDC. In a typical Windows environment, the KDC is part of the DNS record. This might not be the case if Identity Connect is located inside a DMZ.

When IWA is enabled, the session idle timeout function does not work as expected. When a session times out, clicking on the UI triggers a transparent logout and login. This might make it appear as if the session remains active indefinitely.

If your service is behind a load balancer, perform the following additional steps:

1. Link the SPN that was created for the Identity Connect host to the user account that you created for the load balancer.

The following command checks which SPNs are registered for the load balancer user account:

```
C:\Users\Administrator>setspn -L lb-user
Registered ServicePrincipalNames for CN=lb-user,CN=Users,DC=ad,DC=example,DC=com:
HTTP/lb.ad.example.com
```

Currently, only the load balancer SPN is linked.

The following command links the Identity Connect SPN to the load balancer user account. The **-S** option verifies that there are no duplicate SPNs:

```
setspn -S service name/host name user account
```

For example:

```
C:\Users\Administrator> setspn -S HTTP/identityconnect.example.com lb-user
Checking domain DC=ad,DC=example,DC=com
Registering ServicePrincipalNames for CN=lb-user,CN=Users,DC=ad,DC=example,DC=com
HTTP/identityconnect.example.com
Updated object
```

Verify that the new SPN has been registered correctly:

```
C:\Users\Administrator>setspn -L lb-user
Registered ServicePrincipalNames for CN=lb-user,CN=Users,DC=ad,DC=example,DC=com:
HTTP/lb.ad.example.com
HTTP/identityconnect.ad.example.com
```

Both SPNs are now linked to the load balancer user account.

2. If you observe the error **KRB_ERR_RESPONSE_TOO_BIG**, you might need to force Kerberos to use TCP instead of UDP on your Active Directory server.

For more information on how to do this, see the corresponding Microsoft Knowledge Base article.

Identity Connect is now configured for IWA.

In a load-balanced deployment, users would access the server through the load balancer URL (<https://lb.ad.example.com:8443/connect> in our example).

7.4. Configuring Client Browsers for SPNEGO

Identity Connect uses the Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO), to negotiate authentication mechanisms with the client browser. To access their Salesforce accounts through Identity Connect using IWA, clients must use a browser that supports SPNEGO authentication. Most modern browsers support SPNEGO but require some additional configuration to make it work.

The configuration required varies, depending on the operating system of the client from which you will access Identity Connect.

Note

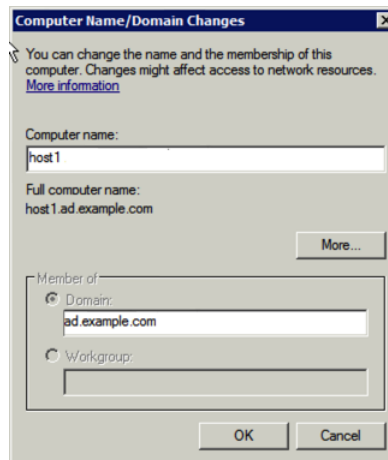
The client and the host on which Identity Connect is installed *must not be the same machine*.

Google Chrome requires no additional configuration to support SPNEGO, except on Mac OS clients. For more information, see Section 7.4.2, "To Enable Kerberos Authentication Mac OS".

7.4.1. Configuring Browsers for SPNEGO on Windows Clients

For Windows clients, the easiest way to configure browsers for SPNEGO is for the Windows client to join the Active Directory domain (ad.example.com in our example). Generally, this is already the case. If your client is not part of the Active Directory domain, follow these steps:

1. From the Control Panel, select System Properties > Advanced > Computer Name > Change.
2. In the *Member of* panel, select *Domain* and enter the name of the Active Directory domain.



To join the domain, you will need to provide the administrator's credentials.

3. After you have joined the domain, reboot your Windows client.

When the Windows client is part of the domain, configure the browser.

7.4.1.1. To Configure Internet Explorer for SPNEGO

1. Launch Internet Explorer and check that the Identity Connect URL is included in the list of "Trusted Sites", as follows:

- From the Tools menu, select Internet Options and select the Security tab.
- Select Trusted Sites and click the Sites button.
- In the list of Websites, check that the URL for Identity Connect appears.
- Click the `Custom Level...` button.
- In the Settings pane, scroll down to User Authentication.

Select `Automatic logon with current username and password`.

2. Select the Advanced tab and scroll down to the list of Security settings.

Make sure that Enable Integrated Windows Authentication is selected.

3. You should now be able to access Identity Connect, seamlessly, through Internet Explorer. No username or password should be required.

Ask your Active Directory administrator to push these changes to all client browsers.

7.4.1.2. To Configure Firefox for SPNEGO

Firefox supports SPNEGO, but it is disabled by default. To enable SPNEGO, follow these steps:

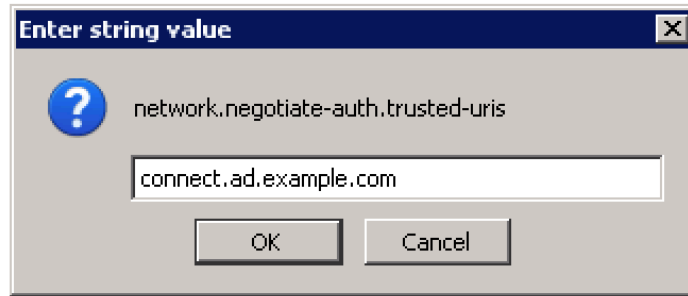
1. Enter the URL `about:config` in the address bar.

Click past the warnings about your warranty.

2. At the top of the page, search for `negotiate-auth` to filter the results.

Double-click on `network.negotiate-auth.trusted-uris`.

3. In the dialog box, enter the Identity Connect URL and click OK.



You should now be able to access Identity Connect, seamlessly, through your Firefox browser, without providing a username and password.

7.4.2. To Enable Kerberos Authentication Mac OS

On a Mac OS client, there are two ways to enable Kerberos authentication.

- Join the Mac OS client to the Active Directory domain.
- Edit the `krb5.conf` to generate tickets by using `kinit`.

7.4.2.1. Joining a Mac OS X Client to an Active Directory Domain

Before you attempt to join an Active Directory domain from a Mac OS client, contact your system administrator, as some of these steps might require specific permissions.

Your Mac must have the following basic networking configuration:

- An IP address and a subnet mask
- A DNS hostname
- A connection to a Windows DNS server

When you join the Mac to the domain, you will need to use the credentials of the domain administrator, or of a user account with the required privileges.

On your Mac client, follow these steps to join the AD domain. These instructions are for Mac OS X Lion. You might need to adjust these instructions for your particular Mac OS version.

1. Select System Preferences > Users and Groups > Login Options
Click the Lock icon to enable you to change these settings.
2. Click the Join button next to Network Account Server.
3. Enter the name of the KDC server (`ad.example.com` in our example)

4. Enter the credentials of the administration user for the KDC server and click OK.

After you have added the Mac to the domain, configure your browser for SPNEGO.

- Safari browsers require no additional configuration.
- For Firefox, edit the list of `network.negotiate-auth.trusted-uris`, as described in Section 7.4.1.2, "To Configure Firefox for SPNEGO".
- Google Chrome requires command line arguments to enable SPNEGO support. To access Identity Connect without a username and password on Chrome, launch Chrome, from a terminal window, as follows:

```
$ open '/Applications/Google Chrome.app' \  
--args \  
--auth-server-whitelist="connect.ad.example.com" \  
--auth-negotiate-delegate-whitelist="connect.ad.example.com" \  
--auth-schemes="digest,ntlm,negotiate" \  
https://connect.ad.example.com:8443/connect
```

7.4.2.2. Using a Kerberos Configuration File to Generate Tickets With **kinit**

If you are unable to join your Mac OS or Unix system to the Windows domain (or if your company policy prevents you from doing so) you can use the **kinit** command to generate Kerberos tickets.

The Kerberos configuration file (`krb5.conf`) contains configuration information that is required by the Kerberos library, including the default Kerberos realm and the location of the Kerberos Key Distribution Center (KDC).

1. Edit the `/etc/krb5.conf` file. (If this file does not exist, create a new `krb5.conf` file in the `/etc` directory.)

The file contents must include the Kerberos information specific to your site, including the permitted encryption types. The following example shows the Kerberos configuration file for the example described previously.

```
$ more /etc/krb5.conf  
[libdefaults]  
default_realm = AD.EXAMPLE.COM  
default_tkt_encypes = arcfour-hmac aes256-cts  
default_tgs_encypes = arcfour-hmac aes256-cts  
  
[realms]  
AD.EXAMPLE.COM = {  
admin_server = 192.0.2.0  
kdc = 192.0.2.0  
kpasswd = 192.0.2.0  
}  
  
[domain_realm]  
.yourdomain.com = AD.EXAMPLE.COM  
localhost = AD.EXAMPLE.COM
```

When the Kerberos configuration file is in place, generate the initial TGT Kerberos ticket that will be used by Safari, Chrome and Firefox to request additional tickets.

2. Use **kinit** to generate the initial ticket:

```
$ kinit admin@AD.EXAMPLE.COM
admin@AD.EXAMPLE.COM's Password: *****
```

The format in which the user name is entered depends on how your client machine is configured (so might be simply **\$ kinit admin** in your case).

3. Run the **klist** command to verify that the ticket has been created.

```
$ klist -v
Credentials cache: API:501:68
Principal: admin@AD.EXAMPLE.COM
Cache version: 0
Server: krbtgt/connect.AD.EXAMPLE.COM@AD.EXAMPLE.COM
Client: admin@AD.EXAMPLE.COM
Ticket etype: aes256-cts-hmac-sha1-96, kvno 2
Ticket length: 1051
Auth time: Jun 4 16:10:43 2013
End time: June 5 02:09:01 2013
Ticket flags: pre-authent, initial, proxiabile, forwardable
Addresses: addressless
```

After the ticket has been generated, launch your browser and edit the list of `network.negotiate-auth.trusted-uris`, as described in the previous section. You should now be able to access Identity Connect through your browser.

Chapter 8

Customizing the Identity Connect Interface

This chapter describes how to customize various aspects of the Identity Connect User Interface.

8.1. Customizing the UI Theme

You can customize various aspects of the login page that users see when they access the Identity Connect user interface (for example, at <https://hostname.domain:8443/connect/>). To change the UI theme, follow these steps:

1. Click the Settings menu at the top right of the Identity Connect administrative interface.
2. Select the Customize Theme tab.
3. Adjust the following settings, as required:

- *Background Color* enables you to set the background color of the login page.

To change the background color, click the Background Color dropdown list. Select the required color from the color panel, or enter the hex value of the required color, and click *Choose*.

- *Button Color* enables you to set the color of all the buttons on the Identity Connect user interface.

To change the button color, click the Button Color dropdown list. Select the required color from the color panel, or enter the hex value of the required color, and click *Choose*.

- *Logo* enables you to select a custom logo to appear on the login screen and in the administrative interface.

To select a custom logo, click Browse, and locate the image file.

4. When you have completed the customization, click Save to save your changes.

8.2. Changing the Password Reset Link

You can reroute password reset in the event that a user has forgotten his password, by specifying an external URL to which password reset requests are sent.

To set an external URL to handle password resets:

1. Click Settings at the top right of the Identity Connect window.
2. Select the Authentication and Session tab.
3. In the *Reset Password Link* field, enter the URL to which password reset requests should be sent.

8.3. Changing the Session Timeout

By default, an Identity Connect UI session times out after 30 minutes of inactivity, or 120 minutes after the initial login, whichever happens first.

To adjust the login session length:

1. Click Settings at the top right of the Identity Connect window.
2. Select the Authentication and Session tab.
3. In the *Maximum session lifetime* field, enter the maximum number of minutes that a session can be live, after the initial login.
4. In the *Session idle timeout* field, enter the maximum number of minutes that a session can be idle, before the user is logged out automatically.

Note

Changes to the session timeout settings only take effect for a new session, that is, you need to log out and log back in to see the effect of the change.

Chapter 9

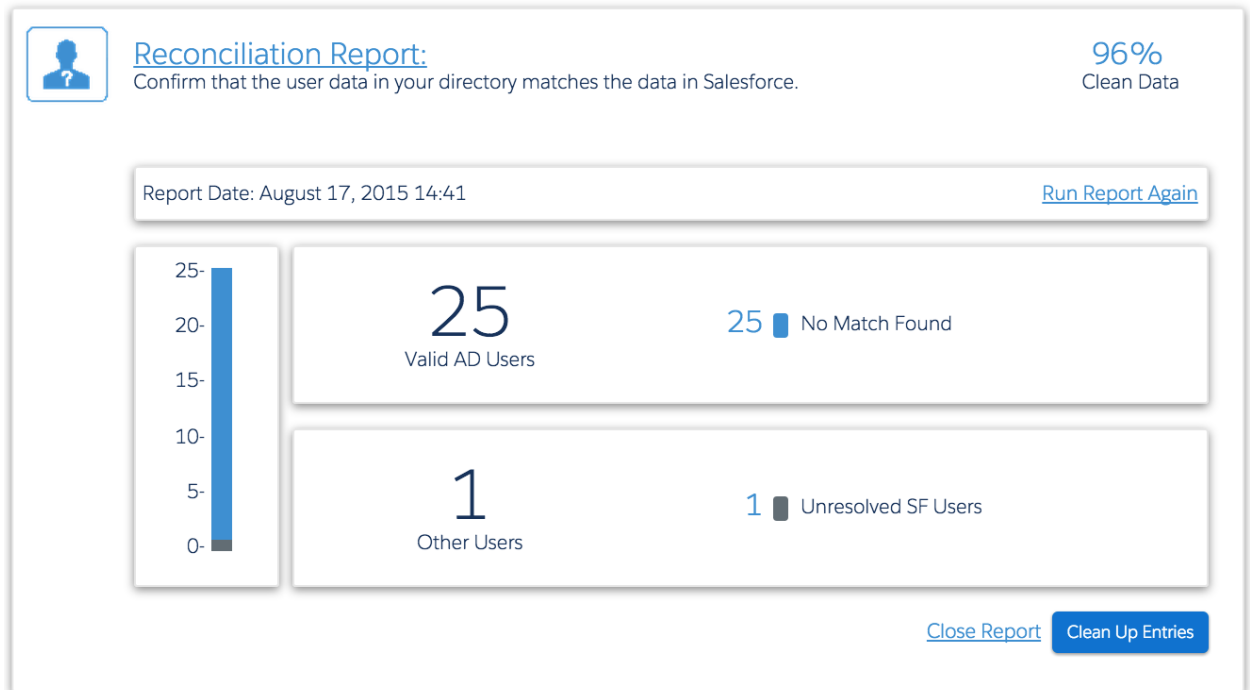
Configuring Auditing and Reporting

This chapter describes how to run reports from Identity Connect to track data consistency, changes to user data, and access to the Identity Connect application itself.

9.1. Running Reconciliation Reports

Reconciliation reports analyze user data to assess the consistency of the data across two data stores. A reconciliation report is generated when you click **Analyze Associations Now** on the Sync page, or when you click **Run Report Again**, on the report itself. Running a reconciliation report enables you to assess the current data consistency. Essentially, a reconciliation report is a synchronization run in *report mode*, where discrepancies in data are reported rather than fixed.

To view the latest reconciliation report, select **Reports** on the Summary page, select the Salesforce organization for which the report should be run, then click **Reconciliation Report**. The fields in the reconciliation report match the user data categorizations described in Chapter 5, "*Data Synchronization and User Association Management*".



To clean up data based on the reconciliation report, click *Clean Up Entries*. After your data has been cleaned, click *Run Report Again* to obtain an updated view of the data consistency.

9.2. Running Synchronization Reports

The Synchronization Report provides details of scheduled and manual synchronization operations and enables you to troubleshoot a failed synchronization.

To view the latest synchronization report, select Reports on the Summary page, select the Salesforce organization for which the report should be run, then click Synchronization Report. Note that this report is displayed only if one reconciliation operation has been launched.

Synchronization Report:

This report displays information about sync success/failure.

Min Date:

Max Date:

Sync Date ↕	Succeeded	Failed
August 17, 2015 14:55	28	1
August 17, 2015 14:54	0	1
August 17, 2015 14:47	1	28

⏪ << Page 1 of 1 >> ⏩

[Export To CSV](#) [Close Report](#)

By default, all synchronization operations are shown, along with the date on which the operation was run, the number of records that were successfully synchronized, and the number of records for which synchronization failed. You can filter the operations to display only those that occurred within a specific date range by providing the Min Date and Max Date and clicking Filter.

For more detail about failed, or successful operations, click the number under the Succeeded or Failed column, corresponding to the date of the synchronization operation that you want to investigate.

In the case of a successful synchronization operation, the additional detail includes the action performed on the record, the record in Active Directory, and its corresponding record in Salesforce. By default, all modifications performed by the synchronization operation are shown, however, you can filter the report based on the action taken. In the following report, only user entries that have been CREATED are displayed.

- Succeeded		
Action	Your Directory	Salesforce
CREATE		
CREATE	Jeannetta Lehman (jlehman)	Jeannetta Lehman (jlehman) jlehman@example.com
CREATE	Bernice Jenner (bjenner)	Bernice Jenner (bjenner) bjenner@example.com
CREATE	Lukhanya Moloi (Imoloi)	Lukhanya Moloi (Imoloi) Imoloi@example.com
CREATE	Kaitlyn Bernstein (kbernste)	Kaitlyn Bernstein (kbernste) kbernstein@example.com
CREATE	Warren Zeeman (wzeeman)	Warren Zeeman (wzeeman) wzeeman@example.com
CREATE	Clive Rice (crice)	Clive Rice (crice) crice@example.com
CREATE	Brian Murray (bmurray)	Brian Murray (bmurray) bmurray@example.com
CREATE	Anna Davis (adavis)	Anna Davis (adavis) adavis@example.com
CREATE	Susan Jenkins (sjenkins)	Susan Jenkins (sjenkins) sjenkins@example.com
CREATE	Michael Hofmeyr (mhofmeyr)	Michael Hofmeyr (mhofmeyr) mhofmeyr@example.com

Page 1 of 3 10

Export To CSV Close

In the case of a failed synchronization operation, the additional detail includes the attempted action, the record in Active Directory, and its corresponding record in Salesforce, and the reason for the failure. You can use this information to correct problematic entries before running the synchronization operation again.

- Failed			
Action	Your Directory	Salesforce	Exception
CREATE	Lukhanya Moloi (Imoloi)	Not Found	Required fields are missing: [Email]

Page 1 of 1 10

Export To CSV Close

To export the report to a CSV file, click Export To CSV at the bottom of the report.

9.2.1. Purging Synchronization Records

By default, synchronization reports are purged every twelve hours, retaining only the latest ten reports. All single user synchronizations are removed from the audit log every time the log is purged.

The purging schedule is configured in two schedule configuration files, located in the `path/to/salesforceIdConnect/conf` directory. Ordinarily, you should not need to modify these files. However, you can modify these schedules if you want to change the purge interval, or the way in which the purging is done.

The schedule that controls purging audit records used for synchronization reporting is configured in the file `salesforceIdConnect/conf/schedule-autoPurgeAuditRecon_ADUsers_SalesForceUsers.json`. The schedule that controls purging audit records for mappings for which the UI needs only one summary record is controlled in the file `salesforceIdConnect/conf/schedule-autoPurgeAuditRecon_ADUsers_SalesForceUsers_Analysis.json`. The contents of these schedule configuration files are similar and are described in this section.

```
{
  "enabled" : true,
  "type" : "cron",
  "schedule" : "0 0 */12 * * ?",
  "persisted" : true,
  "misfirePolicy" : "doNothing",
  "invokeService" : "script",
  "invokeContext" : {
    "script" : {
      "type" : "text/javascript",
      "file" : "script/autoPurgeAuditRecon.js"
    },
    "input" : {
      "mappings" : [
        "ADUsers_SalesForceUsers_%"
      ],
      "purgeByNumSummariesToKeep" : true,
      "numSummaries" : 10,
      "purgeByDate" : false,
      "intervalType" : "days",
      "intervalValue" : 7,
      "scheduleName" : "autoPurgeAuditRecon_ADUsers_SalesForceUsers"
    }
  }
}
```

The purge script (`script/autoPurgeAuditRecon.js`) runs every twelve hours. To change the purge interval, edit the `schedule` property in this file, using standard CRON syntax. For information about CRON syntax, see the Quartz Scheduler documentation.

By default, the purge job retains the latest ten reconciliation summaries, for display in the Synchronization Report. To change the number of summaries that is retained, edit the `numSummaries` property. For example, to specify that the last twenty summaries be retained, and displayed in the Synchronization Report, set `numSummaries" : 20,`

To purge reconciliation summaries according to a date range, rather than the number of summaries to retain, set `"purgeByNumSummariesToKeep" : false`, and `"purgeByDate" : true`. You can then specify the interval for the date range using the `"intervalType"` and `"intervalValue"` properties. The `"intervalType"` can be one of `"minutes"`, `"hours"`, or `"days"`.

Caution

While it is possible to disable purging synchronization audit records, it is highly recommended that you do not do so. When audit records are not purged, the resulting unrestricted disk usage growth can have a significant performance impact on the Identity Connect UI.

After the reconciliation summaries have been purged, you might want to rebuild the synchronization audit indexes, as described in the following section.

9.2.2. Rebuilding Audit Indexes

Synchronization reports are indexed by default. These indexes are stored in tables in the repository, and increase in size over time. From time to time it is necessary to rebuild these audit indexes, to prevent their size from impacting the performance of the Identity Connect user interface.

Depending on the size of your deployment, it is recommended that you rebuild the indexes every couple of weeks to keep the index sizes down. If you are seeing performance issues, you might want to try rebuilding the audit indexes to see if that resolves the problem.

Caution

Rebuilding indexes can cancel any currently running reconciliation operations, so do not start this procedure if reconciliation operations are in progress.

To rebuild the audit indexes, follow these steps:

1. Log into the Identity Connect administrative interface, and click Settings.
2. On the settings page, select the Database tab.
3. At the bottom of the page, click Rebuild Audit Indexes.

9.3. Running User Activity Reports

The User Activity Report provides information about successful and failed login attempts to Identity Connect.

To view the latest Activity Report, select Reports on the Summary page, then click User Activity Report. You can filter the report output by username, IP address, or within a specific date range.

To export the report to a CSV file, click Export To CSV at the bottom of the report.

User Activity Report:

This report displays information about login attempts.

2
Active Logins

Successful Logins

Failed Logins

Username:

Min Date:

IP:

Max Date:

Username	IP	Date/Time ▾	Active
bjensen	127.0.0.1	August 17, 2015 14:10:38	✔
bjensen	127.0.0.1	August 17, 2015 13:57:44	✔

⏪ << Page 1 of 1 >> ⏩

[Export To CSV](#)

[Close Report](#)

Note that logins remain active until they expire, and are based on the validity of the JWT session cookie. Therefore if a specific user logs in, shuts down his browser session, and logs in again, two logins will be active for that user.

Chapter 10

Securing an Identity Connect Deployment

This chapter describes how to manage keys and certificates to establish trust between users and Identity Connect, and provides additional information about securing an Identity Connect deployment.

10.1. Managing SSL Certificates

Identity Connect provides a self-signed certificate for evaluation purposes. In production systems, you should use a certificate that has been signed by a certificate authority to establish trust between users and Identity Connect. A CA-signed certificate will prevent users from seeing the certificate warning when they log on to their Salesforce dashboard via Identity Connect.

The Identity Connect administrative user interface provides a mechanism to generate certificate signing requests, and to import the resulting signed certificates into Identity Connect's keystore. You can also use the Java **keytool** command to manage certificates with the command line interface. Because the **keytool** command does not include the ability to write to a PKCS12 database, this section also provides instructions on using the **openssl** command to work with PKCS12 keystores.

Important

If you use a 2048-bit SSL certificate, you *must* install the Unlimited JCE policy for your JRE to enable Identity Connect to use the certificate.

Download and install the Unlimited JCE Policy for Java 8 from the Oracle Technetwork site. Unzip the JCE zip file and install the JCE policy JAR files in the `/lib/security` folder of the JRE.

The following sections describe how to manage SSL certificates for Identity Connect.

10.1.1. Managing SSL Certificates Through the UI

The following procedure shows how to generate a certificate signing request (CSR) by using the UI, and to import the signed certificate into Identity Connect's keystore.

1. Log into the Identity Connect administration interface, (for example, <https://hostname.domain:8443/admin>).
2. Click Settings in the top right corner and select the SSL Configuration tab.
3. On the SSL Configuration tab, enter the details of the CSR, then click Generate CSR.

Generate certificate signing request (CSR)

Provide necessary details for requesting a new certificate

Common Name/CN (Server Host Name)
 ✓

Organization
 ✓

Organizational Unit
 ✓

City or Town
 ✓

State or Province
 ✓

Country
 ✓

4. The resulting CSR is displayed in a new window. Copy the contents of the CSR and submit it to your Certificate Authority for signing.

Certificate Signing Request ✕

```
-----BEGIN CERTIFICATE REQUEST-----
MIICqjCCAZICAQAwZzESMBAGA1UEAwWJbG9jYWxob3N0MQ4wDAYDVQQQLDAVhZG1p
bjEUMBIGA1UECgwLZXhhbXBsZS5jb20xETAPBgNVBACMCFBvcnRsYW5kMQswCQYD
VQQIDAJPujELMAkGA1UEBhMCVWwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQDvr4P/zyGDpSR18sQp9Oj/9az44Cv04Awly9OQFmovRtrA8MLlci9yCPhn
KMJnxn0cW48fxyGVy5XVI7E0cva8N6pVNIkFVo3S1nc7JRFWNvku8c24ZQMe1LNB
haA/I1EArbMd87GGFtj9dk031GmxcOayOtIpOVP2dMws0M7VKodEj1Kp6HvWkrJF
jVT8ZvxyLUgrAAg705Cldk5x2CLx00SMzoe+/UehE7SL5gzxfnkHh85Gen2+X6r9
JT+QbNP6fyOyZtPVgFE8og1LyhY5zPNGuTKKRG6V9r5YT5g1LyaJqanwITOW55o
pnR8jtlmDFTRZ4QfcCn+j5Iim3FrAgMBAAEwDQYJKoZIhvcNAQENBQADggEBAKGZ
2HHuZ7ctYEO2FM63fESntiyl07FRcUemqspIcUqeUd+wGw9/kTYvtr+RBOs13YPI
+DnX3Z/LzojIqWYuZsIaOqh2mVURTfDeo8xmSiPzPbJnFeYULE6tPGSvI/z+UWhg
cuviMuChCrn5i+KsOnwJaaRop4rrzLf3dT+bdqzxufaE6bLU2vzYRwvdsWiApC
OjU4qmTVcbn/W0E/a30X09x3/z72JKPlus/wj9Cgp5eTAbfweNB3I3JhGzjZH0F/
fm2L0pjdLoLVuOpHAXx0ZN4lDk2qZvEfssaQJW0NSdeWPTtTwrL7Gj3o/9why12yE
OPYZqkpnRwIJGyi6X5s=
-----END CERTIFICATE REQUEST-----
|
```

Copy this value and submit it to your Certificate Authority for processing. When they sign the request, return to the Settings screen and choose "Complete CSR Process".

- When the signed certificate is returned from your certificate authority, click Settings again, select the SSL Configuration tab, and click Complete CSR Process.
- Copy and paste the contents of the CA-signed certificate (PEM file) into the first text box.

Complete CSR process

If you have started a previous CSR process, enter the signed certificate you received from your Certificate Authority (PEM format):

```

-----BEGIN CERTIFICATE-----
MIICGjCAGCAZCAQAwZzESMBAGA1UEAwJbG9jYWxob3N0M04wDAYDVQQLDAVhZG1p
jVt8ZvxyLUgrAAg7O5ClDk5x2CLxO0SMzoe+/UehE7SL5gzzfXkHh85Gen2+X6r9
VQOIDAJPUjELMAkGAlUEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoTBAQDVR4P/zYGDPsSR18sOp9Oj/9aZ44Cv04Awly9OOFmovRtrA8MLlci9yCPHn
KMJxn0cW48fxyGVy5XVI7E0cva8N6pVNIkFVo3S1nc7JRFWNvku8c24ZQMe1LNB
haA/1lEArBmd87GGFtj9dk031GmxcOayOtIpOVP2dMws0M7VKodEj1Kp6HvWKRJF
jVt8ZvxyLUgrAAg7O5ClDk5x2CLxO0SMzoe+/UehE7SL5gzzfXkHh85Gen2+X6r9
JT+QbNP6fyOyZtPVqFE8og1LyhY5zPngUTKKRG6V9r5YT5g1LyaJjganwITOW55o
pnR8itlmdFTRZ4QfcCn+i5Im3FrAgMBAAEFwDOYJKoZlhwNAOENBQADggEBBAKGZ
2HHuZ7ctYE02FM63fESntiylo7FRcUemqspIcUgeUd+wGw9/kTYvtR+RBOS13YPi
+DnX3Z/LzqjIqWYuZsIaOqh2mVURTfDeo8xmSiPzPbJnFeYULe6tPGSyI/z+UWhg
cuvimuChbCn5i+KsOnwJaaRQp4rrzLf3dT+bdogzxufaE6blU2vzYRwvdsWiAoC
QjU4qmTvcbn/W0E/a30X09x3/z72JKPlus/wj9Cqp5eTAbfweNB3I3JhGzjzH0F/
fm2L0pjdLoLVuOpHAXx0ZN41Dk2qZvEfsaaQJWONSdeWPtTwrL7Gj3o/9why12vE
jVt8ZvxyLUgrAAg7O5ClDk5x2CLxO0SMzoe+/UehE7SL5gzzfXkHh85Gen2+X6r9
VQOIDAJPUjELMAkGAlUEBhMCVVMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
KMJxn0cW48fxyGVy5XVI7E0cva8N6pVNIkFVo3S1nc7JRFWNvku8c24ZQMe1LNB
2HHuZ7ctYE02FM63fESntiylo7FRcUemqspIcUgeUd+wGw9/kTYvtR+RBOS13YPi
0PYZqkpnRwIJGyi6X5s=
-----END CERTIFICATE-----
    
```

Note: you may need to restart your browser after uploading the new certificate, as the old certificate may be cached.

- If your CA has provided an intermediate or root certificate, click Include Additional Certificates and copy and paste the contents of those certificates.
- Click Upload Signed Certificate to import the CA certificate and the corresponding certificate chain into the keystore.
- When you have imported the CA certificate into Identity Connect's keystore, restart Identity Connect for the new certificate to be taken into account.
- Restart your browser after you have uploaded the new certificates, because the old self-signed certificate might be cached.

Note

The HTTPS server certificate is also used for LDAPS client authentication when requested by the Active Directory server. If you are connecting to Active Directory over LDAPS on a Microsoft Windows 2012 R2 server, you *must* import a CA-signed certificate into Identity Connect, as described in the previous steps. You should never use a self-signed certificate in a production environment.

10.1.2. Managing SSL Certificates by Using **keytool**

The following procedure shows how to generate a certificate signing request by using the command line. This procedure uses the private key that is provided with Identity Connect. You can also use an existing private key, or create a new private key for this certificate.

For more information about the **keytool** command, see the documentation at <https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html>.

To generate a CSR for Identity Connect using **keytool**, follow these steps:

1. Delete the default keystore alias (**openidm-localhost**).

```
$ keytool \  
-delete \  
-alias openidm-localhost \  
-keystore /path/to/salesforceIdConnect/security/keystore.jceks \  
-storetype JCEKS  
Enter keystore password: changeit
```

2. Create a new private key pair that will be used for the certificate signing request (CSR), using the same default alias.

```
$ keytool \  
-genkey \  
-keysize 2048 \  
-keyalg RSA \  
-sigalg SHA1withRSA \  
-alias openidm-localhost \  
-keystore /path/to/salesforceIdConnect/security/keystore.jceks \  
-storetype JCEKS \  
-storepass changeit \  
-dname "CN=connect.example.com,O=Example,L=Portland,C=US"  
Enter key password for <openidm-localhost>  
(RETURN if same as keystore password):
```

3. Generate a certificate signing request (CSR) with the private that key you created in the previous step. Save the CSR in a file named **connect.csr**.

```
$ keytool \  
-certreq \  
-alias openidm-localhost \  
-keystore /path/to/salesforceIdConnect/security/keystore.jceks \  
-storetype JCEKS \  
-file connect.csr  
Enter keystore password: changeit
```

4. Submit the CSR to a certificate authority for signing.
5. When the signed certificate and the root CA certificate are returned by the CA, import the root CA certificate into the keystore first. Substitute `ca-certificate.pem` with the name of your CA certificate file, and select to trust the certificate when prompted.

```
$ keytool \  
-import \  
-trustcacerts \  
-alias CARoot \  
-file ca-certificate.pem \  
-keystore /path/to/salesforceIdConnect/security/keystore.jceks \  
-storetype JCEKS  
Enter keystore password: changeit
```

6. After you have imported the root CA certificate, import the signed certificate into the keystore, as follows:

```
$ keytool \  
-import \  
-trustcacerts \  
-alias openidm-localhost \  
-file connect.crt \  
-keystore /path/to/salesforceIdConnect/security/keystore.jceks \  
-storetype JCEKS  
Enter keystore password: changeit
```

10.1.3. Importing an Existing Signed Certificate

If you already have a CA-signed certificate, and do not need to create a CSR, you can import that certificate into Identity Connect's keystore by using the command line. You cannot import an existing certificate into the keystore by using the UI.

The following procedure imports an existing PKCS12 wildcard certificate, and its private key, into Identity Connect's keystore. The procedure is the same, whether the certificate you are importing is a wildcard certificate or a more restrictive certificate.

1. View the contents of the keystore.

After startup, the Identity Connect keystore contains two entries. You can see the existing entries by sending the following request to the `security` endpoint. The default Identity Connect keystore password is `changeit`.

```
$ cd /path/to/salesforceIdConnect/security
$ keytool \
  -list \
  -keystore keystore.jceks \
  -storetype JCEKS
Enter keystore password:changeit

Keystore type: JCEKS
Keystore provider: SunJCE

Your keystore contains 2 entries

openidm-sym-default, 20 Aug 2015, SecretKeyEntry,
openidm-localhost, 20 Aug 2015, PrivateKeyEntry,
Certificate fingerprint (SHA1): 9A:D6:9E:...D6:81:D8:21
```

Identity Connect uses the certificate with the alias `openidm-localhost`, by default.

2. Delete the default certificate alias (`openidm-localhost`) from the keystore.

```
$ keytool \
  -delete \
  -alias openidm-localhost \
  -keystore keystore.jceks \
  -storetype JCEKS
Enter keystore password: changeit
```

3. Use the **keytool** command to import your existing certificate into the keystore.

Substitute `example-cert.p12` with the name of your certificate file and `changeit` with the password that you set to open your certificate. This command assumes that the existing certificate alias is `example-com`. The certificate will be imported into the keystore with the alias `openidm-localhost`. The **keytool** command creates a trusted certificate entry with the specified alias and associates it with the imported certificate.

```
$ keytool \
  -importkeystore \
  -srckeystore example-cert.p12 \
  -srcstoretype PKCS12 \
  -srcstorepass changeit \
  -alias example-com \
  -destkeystore keystore.jceks \
  -deststoretype JCEKS \
  -destalias openidm-localhost
Enter keystore password: changeit
```

Note

Identity Connect requires that the certificate password and keystore passwords are the same. If the password of your existing certificate, is not the same as the Identity Connect keystore password, change its password to match that of the keystore, as follows:

```
$ keytool \  
-keypasswd \  
-alias openidm-localhost \  
-keystore keystore.jceks \  
-storetype JCEKS  
Enter keystore password: changeit  
Enter key password for <openidm-localhost>old-password  
New key password for <openidm-localhost>: changeit  
Re-enter new key password for <openidm-localhost>: changeit
```

4. When you have imported the certificate, view the contents of the keystore again, to verify that your certificate is there:

```
$ keytool \  
-list \  
-keystore keystore.jceks \  
-storetype JCEKS  
  
Enter keystore password:changeit  
  
Keystore type: JCEKS  
Keystore provider: SunJCE  
  
Your keystore contains 2 entries  
  
openidm-sym-default, 20 Aug 2015, SecretKeyEntry,  
openidm-localhost, 20 Aug 2015, PrivateKeyEntry,  
Certificate fingerprint (SHA1): 8C:A2:21:...ED:15:C2:C8:22:C0:1E
```

10.2. Configuring Identity Connect for Client Certificate Authentication

By default, client certificate authentication is disabled in Identity Connect. If you want to use mutual authentication, you must adjust the web server SSL settings to enable client certificate authentication.

To enable client certificate authentication, edit the `/path/to/salesforceIdConnect/conf/jetty.xml` file as follows:

```
<Set name="wantClientAuth">true</Set>  
<Set name="needClientAuth">true</Set>
```

10.3. Obfuscating Bootstrap Information

Identity Connect uses the information in `/path/to/salesforceIdConnect/conf/boot/boot.properties`, including the key store password, to start up. The key store password is `changeit` by default, and is stored in clear text in the `boot.properties` file. To set an obfuscated version of the key store password in the `boot.properties` file, follow these steps.

1. Generate an obfuscated version of the password, by using the crypto bundle that is provided with Identity Connect:

```
$ java -jar /path/to/salesforceIdConnect/bundle/openidm-crypto-2.1.0-IC-1.0.5.jar
This utility helps obfuscate passwords to prevent casual observation.
It is not securely encrypted and needs further measures to prevent disclosure.
Please enter the password:
OBF:1vn2lugu1saj1v9ilv941sarlugw1vo0
CRYPT:a8b5a01ba48a306f300b62a1541734c7
```

2. Paste either the obfuscated password (`OBF:xxxxxxx`) or the cryptographic key (`CRYPT:xxxxxxx`) into the `conf/boot/boot.properties` file. Comment out the regular key store password and remove the comment tags from the line that contains the obfuscated password or cryptographic key, depending on which one you have used:

```
$ more conf/boot/boot.properties
...
# Keystore password, adjust to match your keystore and protect this file
# openidm.keystore.password=changeit
openidm.truststore.password=changeit

# optionally use the cli encrypt to obfuscate the password and set
# openidm.keystore.password=OBF:1vn2lugu1saj1v9ilv941sarlugw1vo0
openidm.keystore
.password=CRYPT:a8b5a01ba48a306f300b62a1541734c7
...
```

3. Restart Identity Connect.

```
$ cd /path/to/salesforceIdConnect
$ nohup ./startup.sh > logs/console.out 2>&1&
[1] 32548
```

Chapter 11

Installing an Alternative Repository

By default, Identity Connect stores its configuration in an internal OrientDB repository. (User entries are not stored in the internal repository.) In certain situations, you might want to use your own SQL database to store the server configuration (for example, if you are setting up a clustered Identity Connect deployment, or if you have an existing SQL database that you would prefer to use).

Identity Connect supports the use of MySQL and MS SQL Server as a repository. For details of the supported versions, see *Supported Repositories* in the *Identity Connect 2.1.0 Release Notes* in the *Release Notes*.

This chapter describes the steps required to get Identity Connect up and running with either MySQL or MS SQL Server as a repository.

11.1. Setting Up Identity Connect With MySQL

Set up Identity Connect to use a MySQL repository, as described in the following procedure. This procedure assumes that:

- Identity Connect has been downloaded and unzipped *but not configured*, that is, the setup process has not been run.
- MySQL has been installed, either on the host on which Identity Connect will run, or on a host that is accessible to the Identity Connect instance.

Procedure 11.1. To Set Up Identity Connect With MySQL

1. Download MySQL Connector/J, version 5.1 or later from the MySQL website. Unpack the delivery, and copy the .jar into the `salesforceIdConnect/bundle` directory.

```
$ cp mysql-connector-java-version-bin.jar /path/to/salesforceIdConnect/bundle/
```

2. Remove the default OrientDB configuration file (`/path/to/salesforceIdConnect/conf/repo.orientdb.json`) from the configuration.

```
$ cd /path/to/salesforceIdConnect/conf/  
$ rm repo.orientdb.json
```

3. Copy the MySQL JDBC configuration file (`/path/to/salesforceIdConnect/db/scripts/mysql/repo.jdbc-mysql.json`) to the `salesforceIdConnect/conf` directory and rename it `repo.jdbc.json`.


```
$ cd /path/to/salesforceIdConnect/conf
$ cp ../db/scripts/mysql/repo.jdbc-mysql.json repo.jdbc.json
```

4. Import the data definition language script for Identity Connect into MySQL.

```
$ cd /path/to/mysql
$ ./bin/mysql -u root -p < \
/path/to/salesforceIdConnect/db/scripts/mysql/openidm_SalesforceIdentityConnect-MySQL.sql
Enter password:
$
```

Enter the root user password for the MySQL server.

This step creates a database named `openidm` for use as the internal repository, and a user `openidm` with password `openidm` who has all the required privileges to update the database.

Load the `openidm` database and verify that you can display the default Identity Connect tables.

```
$ cd /path/to/mysql
$ ./bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.5.19 MySQL Community Server (GPL)
...
mysql> use openidm;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_openidm |
+-----+
| auditaccess        |
| auditactivity      |
| auditrecon         |
| clusterobjectproperties |
| clusterobjects    |
| configobjectproperties |
| configobjects     |
| genericobjectproperties |
| genericobjects    |
| internaluser       |
| links              |
| managedobjectproperties |
| managedobjects    |
| objecttypes        |
| orphanarium        |
| orphanariumproperties |
| schedulerobjectproperties |
| schedulerobjects  |
| uinotification     |
+-----+
19 rows in set (0.00 sec)
```

The table names are similar to those used with the embedded OrientDB repository.

5. Update `salesforceIdConnect/conf/repo.jdbc.json` as necessary, to reflect the details of your MySQL deployment.

```
"connection" : {
  "dbType" : "MYSQL",
  "jndiName" : "",
  "driverClass" : "com.mysql.jdbc.Driver",
  "jdbcUrl" : "jdbc:mysql://localhost:3306/openidm?characterEncoding=utf8",
  "username" : "openidm",
  "password" : "openidm",
  "defaultCatalog" : "openidm",
  "maxBatchSize" : 100,
  "maxTxRetry" : 5,
  "enableConnectionPool" : true,
  "connectionTimeoutInMs" : 30000
},
```

Specifically, make sure that the `username` and `password` used to access the database are correct, and that the `jdbcUrl` reflects the location of the database. The MySQL database can be either local (that is, running on the same host as the Identity Connect instance) or remote (running on a different host to the Identity Connect instance). If the database is on a remote host, adjust the `jdbcUrl` property accordingly, for example `"jdbcUrl" : "jdbc:mysql://remoteMySQLServer.domain.com:3306/openidm?characterEncoding=utf8"`.

6. For optimum performance, tune the MySQL database to work effectively with Identity Connect. The following tuning steps are recommended:
 - Ensure that at least 4GBytes of memory is available to the MySQL instance.
 - Set the `innodb_buffer_pool_size` variable to at least 2GBytes. The value of this variable will depend on the amount of memory that is available to the MySQL instance. As a rule of thumb, the buffer pool size should be approximately 70%-80% of the memory available to the MySQL instance. For more information about this variable, see the corresponding MySQL documentation.
 - Set the `innodb_file_per_table` variable to ON. For more information about this variable, see the corresponding MySQL documentation.
 - On UNIX-like systems only, set the `innodb_flush_method` variable to `O_DIRECT`. For more information about this variable, see the corresponding MySQL documentation.
 - Start the Event Scheduler by setting the `event_scheduler` variable to ON. For more information about this variable, see the corresponding MySQL documentation.

Caution

Enabling the Event Scheduler is *required*. If you do not enable the Event Scheduler, the maintenance process that is used to keep the size of the log data to a minimum cannot run.

- Set the transaction isolation level to **READ-COMMITTED**. For more information about setting the transaction isolation level, see the corresponding MySQL documentation.

When you have set up MySQL for use as the Identity Connect internal repository, start Identity Connect and check the output log (`logs/console.out`), to make sure that the startup has been successful.

```
$ cd /path/to/salesforceIdConnect
$ nohup ./startup.sh > logs/console.out 2>&1&
$ tail -f logs/console.out
Executing ./startup.sh...
Using OPENIDM_HOME: /path/to/salesforceIdConnect
Using OPENIDM_OPTS: -Xmx2048m -Xms2048m
Using LOGGING_CONFIG:
-Djava.util.logging.config.file=/path/to/salesforceIdConnect/conf/logging.properties
Using boot properties at /path/to/salesforceIdConnect/conf/boot/boot.properties
->
```

Log in to the Identity Connect administration console (<https://hostname.domain:8443/admin/>) to confirm that you can access the UI and continue the configuration with the MySQL repository.

11.2. Setting Up Identity Connect With MS SQL Server

Set up Identity Connect to use an MS SQL Server repository, as described in the following procedure. This procedure assumes that:

- Identity Connect has been downloaded and unzipped *but not configured*, that is, the setup process has not been run.
- MS SQL Server has been installed, either on the host on which Identity Connect will run, or on a host that is accessible to the Identity Connect instance.

These instructions are specific to MS SQL Server 2012 R2 Standard Edition, running on a Windows Server 2012 R2 system. Adapt the instructions for your environment.

When you install MS SQL Server, note that Identity Connect has the following specific configuration requirements:

- During the Feature Selection installation step, make sure that at least SQL Server Replication, Full Text Search, and Management Tools - Basic are selected.

These instructions require SQL Management Studio so make sure that you include Management Tools in the installation.

- During the Database Engine Configuration step, select Mixed Mode (SQL Server authentication and Windows authentication). Identity Connect *requires* SQL Server authentication.
- TCP/IP must be enabled and configured for the correct IP address and port. To configure TCP/IP, follow these steps:
 1. Navigate to SQL Server Configuration Manager.
 2. Expand the SQL Server Network Configuration item and select "Protocols for MSSQLSERVER".
 3. Check that TCP/IP is Enabled.
 4. Select the IP Addresses tab and set the addresses and ports on which the server will listen.

For this sample procedure, scroll down to IPAll and set TCP Dynamic Ports to 1433 (the default port for MS SQL).

5. Click OK.
6. Restart MS SQL Server for the configuration changes to take effect.

To restart the server, select SQL Server Services in the left pane, double click SQL Server (MSSQLSERVER) and click Restart.

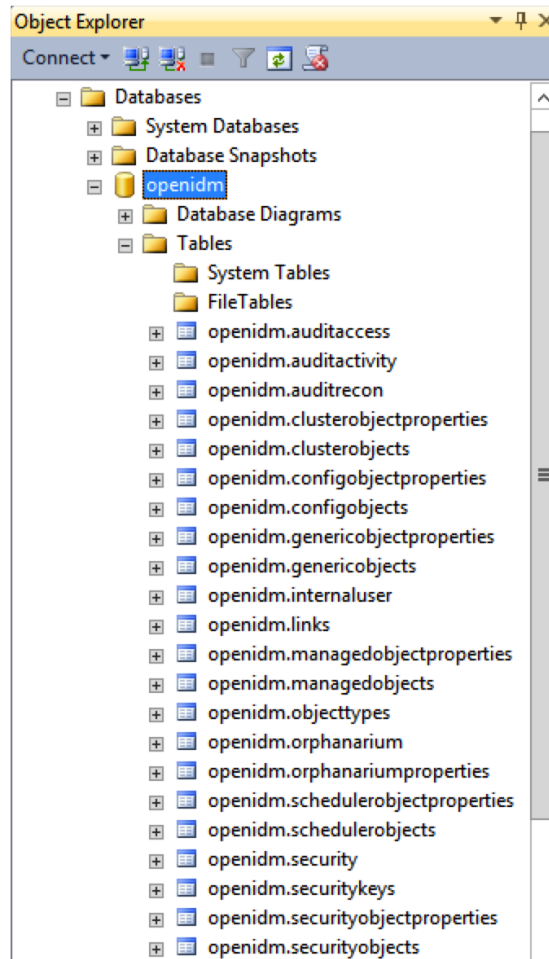
7. If you have a firewall enabled, ensure that the port you configured in the previous step is open for Identity Connect to access MS SQL.

After you have installed MS SQL on the local host import the data definition and set up Identity Connect to use the new repository, as described in the following steps:

Procedure 11.2. To Set Up Identity Connect With MS SQL Server

1. Use SQL Management Studio to import the data definition language script for Identity Connect into MS SQL.
 - a. Navigate to SQL Server Management Studio.
 - b. On the Connect to Server panel, select Windows Authentication and click Connect.
 - c. Select File > Open > File and navigate to the Identity Connect data definition language script (`path\to\salesforceIdConnect\db\scripts\mssql\openidm_SalesforceIdentityConnect-MSSQL.sql`). Click Open to open the file.
 - d. Click Execute to run the script.
2. This step creates an `openidm` database for use as the internal repository, and a user `openidm` with password `Passw0rd` who has all the required privileges to update the database. You might need to refresh the view in SQL Server Management Studio to see the `openidm` database in the Object Explorer.

Expand Databases > openidm > Tables. You should see the following tables in the openidm database:



The table names are similar to those used with an OrientDB repository.

Caution

In a production environment, you *should* change the username and password for the user that access the openidm database. To do so, update the `openidm_SalesforceIdentityConnect-MSSQL.sql` script, editing the following line:

```
IF (NOT EXISTS (select loginname from master.dbo.syslogins where
    name = N'your=new-name' and dbname = N'openidm'))
CREATE LOGIN [openidm] WITH PASSWORD=N'your-new-password',
    DEFAULT_DATABASE=[openidm], CHECK_EXPIRATION=OFF, CHECK_POLICY=OFF
GO
```

Update the repository configuration file (`conf/repo.jdbc.json`) to match these details, as described in Step 6.

3. Identity Connect requires an MS SQL driver that must be created from two separate jar files. Create the driver as follows.
 - a. Download the Microsoft JDBC Driver for SQL Server that corresponds to your JVM version from Microsoft's download site. The precise URL might vary, depending on your location.

For Java 8, you can use either version 4.1 or 4.2 of the JDBC Driver.

Extract the executable Java archive file (`sqljdbc41.jar` or `sqljdbc42.jar`) from the zip file, using 7-zip or an equivalent file management application.

Copy the JAR file to `salesforceIdConnect\db\scripts\mssql`.

The remaining steps assume that you have downloaded the 4.1 version of the driver. If you are using version 4.2 of the driver, adjust the instructions accordingly.
 - b. Download the `bnd` Java archive file (`bnd-1.50.0.jar`) that enables you to create OSGi bundles. For more information about `bnd`, see <http://bnd.bndtools.org/>.

Copy the file to `salesforceIdConnect\db\scripts\mssql`.
 - c. Your `salesforceIdConnect\db\scripts\mssql` directory should now contain the following files:

```
PS C:\> ls .\salesforceIdConnect\db\scripts\mssql
Directory: C:\salesforceIdConnect\db\scripts\mssql

Mode                LastWriteTime         Length Name
----                -
-
-a---              7/22/2015   4:18 AM         828249 bnd-1.50.0
.jar
-a---              7/21/2015   6:39 AM         21131  openidm_SalesforceIdentityConnect-MSSQL
.sql
-a---              7/21/2015   6:39 AM         19568  repo.jdbc-mssql
.json
-a---              7/21/2015   6:39 AM           642  sqljdbc4
.bnd
-a---              12/9/2014   2:17 PM        586192  sqljdbc41.jar
```

- d. Change to the `salesforceIdConnect\db\scripts\mssql` directory, and bundle the two jar files together with the following command:

```
PS C:\salesforceIdConnect\db\scripts\mssql> java -jar bnd-1.50.0.jar ^
  wrap -properties sqljdbc4.bnd sqljdbc41.jar
sqljdbc41 236 0
```

Note

This command assumes that you have set the `JAVA_HOME` environment variable to point to a valid JRE installation directory.

This step creates a single `.bar` file, named `sqljdbc41.bar`.

- e. Rename the `sqljdbc41.bar` file to `sqljdbc41-osgi.jar` and copy it to the `salesforceIdConnect\bundle` directory.

```
PS C:\salesforceIdConnect\db\scripts\mssql> mv sqljdbc41.bar sqljdbc41-osgi.jar
PS C:\salesforceIdConnect\db\scripts\mssql> cp sqljdbc41-osgi.jar ..\..\..\bundle
```

4. Remove the default OrientDB repository configuration file (`salesforceIdConnect\conf\repo.orientdb.json`) from the configuration directory.

```
PS C:\> cd salesforceIdConnect\conf
PS C:\> rm repo.orientdb.json
```

5. Copy the MS SQL JDBC configuration file (`salesforceIdConnect/db/scripts/mssql/repo.jdbc-mssql.json`) to the `salesforceIdConnect/conf` directory and rename it `repo.jdbc.json`.

```
PS C:\> cd salesforceIdConnect\db\scripts\mssql
.\> cp repo.jdbc-mssql.json ..\..\..\conf\repo.jdbc.json
```

6. Update the repository configuration file (`repo.jdbc.json` as necessary, to reflect your MS SQL deployment.

```
{
  "connection" : {
    "dbType" : "SQLSERVER",
    "jndiName" : "",
    "driverClass" : "com.microsoft.sqlserver.jdbc.SQLServerDriver",
    "jdbcUrl" : "jdbc:sqlserver://localhost:1433;instanceName=default;
      databaseName=openidm;applicationName=OpenIDM",
    "username" : "openidm",
    "password" : "PasswOrd",
    "defaultCatalog" : "openidm",
    "maxBatchSize" : 100,
    "maxTxRetry" : 5,
    "enableConnectionPool" : true
  },
  ...
}
```

Specifically, check that the port matches what you have configured for MS SQL.

When you have set up MS SQL for use as the Identity Connect repository, set up Identity Connect, as described in Procedure 2.2, "To Install Identity Connect on Windows Systems".

Log in to the Identity Connect administration console (<https://hostname.domain:8443/admin/>) to confirm that you can access the UI and continue the configuration with the MS SQL repository.

Chapter 12

Deploying Identity Connect for High Availability

To ensure availability of the service, you can deploy multiple Identity Connect instances. In a highly available configuration, only the *primary* instance includes a database. Additional *secondary* instances point to the database of the primary instance. The secondary instances maintain a cached copy of the configuration, and of the list of ignored users, in memory. For more information about ignored users, see Section 5.1, "Overview of the Synchronization Process".

Each secondary instance also contains its own local keystore. The required security certificates are copied into the keystore of the secondary instance from the shared (primary instance) repository when the secondary instance is first brought online.

In the event of the primary instance failing, the secondary instances continue to serve requests until the primary instance comes back online.

Specific configuration changes must be made to configure multiple instances that use a shared repository. These configuration changes are described in this chapter.

In Identity Connect 2.1.0, MySQL is the only supported repository for use in a clustered environment. There are known limitations with the use of the embedded OrientDB repository in a clustered Identity Connect deployment in this release. For information about setting up a MySQL repository, see Chapter 11, "*Installing an Alternative Repository*".

Important

Pass-through authentication to Active Directory cannot function if the connection to the backend database is lost. For a truly highly available deployment, you must also configure the backend database for high availability.

12.1. Configuring High Availability With MySQL

This procedure describes how to configure multiple Identity Connect instances for high availability, when you are using an external MySQL database.

1. On each host that will contain an Identity Connect instance, unpack the contents of the .zip file into the install location, but do not set up Identity Connect.
2. Configure each Identity Connect instance for a remote MySQL database, as described in Chapter 11, "*Installing an Alternative Repository*".

Note that you only need to import the data definition language script (Step 4 of this procedure) for the primary Identity Connect instance. Additional instances will read the database from the primary instance.

When you edit the `/path/to/salesforceIdConnect/conf/repo.jdbc.json` file (Step 5 of this procedure), set the `"jdbcUrl"` property to point to the remote MySQL server. For example:

```
"jdbcUrl" : "jdbc:mysql://server-ip:3306/openidm?characterEncoding=utf8"
```

where `server-ip` is the IP address of the server on which the MySQL server is located.

3. On each Identity Connect instance, edit the `conf/boot/boot.properties` file, as follows:
 - a. Specify a unique identifier for the instance.

For example, on the primary instance:

```
$ grep openidm.node.id /path/to/salesforceIdConnect/conf/boot/boot.properties
openidm.node.id=IdentityConnect1
```

On subsequent instances, the `openidm.node.id` can be set to `IdentityConnect2`, `IdentityConnect3`, and so forth. You can choose any value, as long as they are unique.

- b. Specify the instance type in the cluster.

On the primary instance, add the following line to the `boot.properties` file:

```
openidm.instance.type=clustered-first
```

On subsequent instances, add the following line to the `boot.properties` file:

```
openidm.instance.type=clustered-additional
```

4. On all instances except the primary instance, edit the `conf/system.properties` file, to prevent Identity Connect from reading its configuration from the configuration files, by uncommenting the line `openidm.fileinstall.enabled=false`.

This forces Identity Connect to read its configuration only from the repository, in this case, the repository of the primary instance.

```
$ grep openidm.fileinstall /path/to/salesforceIdConnect/conf/system.properties
openidm.fileinstall.enabled=false
```

5. Start up the primary Identity Connect instance and configure it.

Make sure that when you initially access this Identity Connect instance, you use the FQDN (`https://host.domain:8443/admin`) and not `localhost`.

6. If you imported a certificate for Active Directory during the Identity Connect setup, or if you have modified the truststore on the primary instance, copy the `truststore` file from the primary instance

to the secondary (and additional) instances. For example, run the following command *on the primary instance*, substituting the hostname or IP address of each secondary instance:

```
$ cd /path/to/salesforceIdConnect
$ scp security/truststore admin@host2.example.com:/home/testuser/salesforceIdConnect/security/
```

7. Make sure that the secondary instance (and any additional instances) have access to the MySQL database on the primary instance.

You can check that the secondary instance has access by running a command similar to the following:

```
mysql -u openidm -p -h primary.instance.host.name openidm
```

If your secondary instance cannot access the primary database, you might need to set the appropriate privileges. For example:

```
GRANT ALL PRIVILEGES ON openidm.* to 'openidm'@'primary.instance.host.name' IDENTIFIED BY 'openidm';
```

To identify the privileges required on the MySQL server, run the following command:

```
select user,host from mysql.user;
```

This command displays the user and the related host, and all the privileges that are granted to that user.

8. Start up the secondary (and additional) instances.

Additional instances read the configuration from the first instance, so requests to <https://host2.example.com:8443/connect> should read the existing Identity Connect configuration from the first host.

12.2. Configuring a Load Balancer

After you have configured multiple Identity Connect instances to work together in a cluster, you can configure a load balancer of your choice to distribute client connections between the instances.

Identity Connect 2.1.0 was tested with Nginx Version 1.2.9, but any load balancer that supports sticky session configuration should be adequate.

If you configure a load balancer for Identity Connect, you must specify that the logout from Salesforce be directed to the load balancer. To specify the logout from Salesforce:

1. Log into your Salesforce organization and enter the organization Setup.
2. Under the Administer section, select Single Sign-On Settings from the Security Controls menu.
3. Edit the SAML Single Sign-On Settings to indicate the hostname and port number of the load balancer in the Identity Provider Logout URL field.

Note

The load balancer should not send clients to the secondary hosts for administration or configuration of Identity Connect. Configuration should be handled only by the primary host. Therefore, <https://host2.example.com:8443/admin>, for example, should not be accessible through the load balancer.

If you are configuring IWA for an Identity Connect service behind a load balancer, make sure that the load balanced deployment works with the SAML login first. To do so, configure Identity Connect using the load balancer URL, with the load balancer pointing to the primary server. For more information about configuring IWA for a load balanced deployment, see Section 7.2.3, "Setting Up The Keytab File For a Load Balanced Deployment".

12.3. Configuration Changes in a Clustered Environment

In a clustered environment, any configuration changes must be applied in a specific order. You *must* change the primary node first, then restart all additional (secondary) nodes so that the configuration change is propagated to the secondary nodes.

When you upgrade Identity Connect, the upgrade, and any subsequent configuration changes, must also follow this order. For more information, see Section 2.6, "Upgrading an Identity Connect Instance".

Chapter 13

Advanced Configuration

This chapter provides additional information about the Identity Connect setup. The information in this chapter is not required for you to be able to get Identity Connect up and running, but might help you to understand some of the lower level configuration, and might provide some assistance when troubleshooting an installation. Additional troubleshooting information is provided in Chapter 14, "*Troubleshooting an Identity Connect Installation*".

13.1. Managing the Internal Repository

Identity Connect is provided with an internal noSQL database, OrientDB, for use as the internal repository out of the box. If you use the OrientDB repository, and do not specify an external repository, the following administrative instructions might be useful.

For information about configuring an external repository, see Chapter 11, "*Installing an Alternative Repository*".

13.1.1. Querying the Internal Repository

If you want to query the internal noSQL database, you can download OrientDB (version 1.6.4) from <http://orientdb.com/download-previous/>. You will find the shell console in the `bin` directory. Start OrientDB console using either `console.sh` or `console.bat`, and then connect to the running Identity Connect instance, with the `connect` command. The default Identity Connect database name is `openidm` and the default username and password are `admin` and `admin`.

```
$ /path/to/orientdb-community-1.6.4/bin/console.sh
OrientDB console v.1.6.4 (build @BUILD@) www.orienttechnologies.com
Type 'help' to display all the commands supported.

Installing extensions for GREMLIN language v.2.5.0-SNAPSHOT

orientdb> connect remote:localhost/openidm admin admin
Connecting to database [remote:localhost/openidm] with user 'admin'...OK

orientdb>
```

When you have connected to the database, you might find the following commands useful.

info

Shows classes and records

select * from managed_group

Shows the groups configured in the Active Directory.

select * from audit_recon

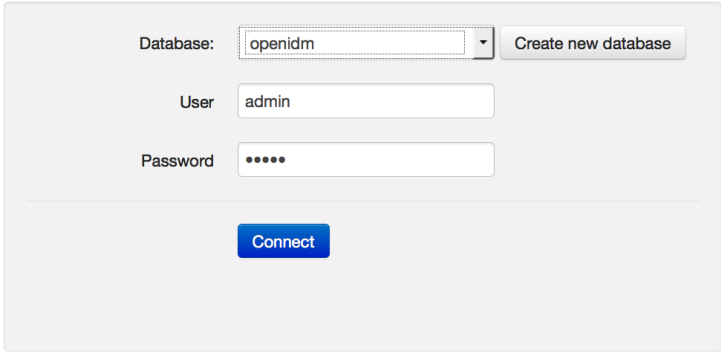
Shows all reconciliation audit records

This table is created when you run reconciliation.

You can also use OrientDB Studio to query the OrientDB repository. OrientDB studio is provided for debugging purposes and is enabled by default. After you have installed and started Identity Connect, access OrientDB Studio as follows:

1. Point your browser to <http://localhost:2480/> and enter your authentication details in the Database access pane.

Database access



Database:

User:

Password:

The default database name for Identity Connect is `openidm`. The default administrator username and password are `admin` and `admin` respectively. The admin user has unrestricted access to all database functions.

2. Click Connect to connect to the repository.

For more information about OrientDB and OrientDB Studio, see the [OrientDB Studio documentation](#).

Caution

OrientDB Studio is required by the maintenance process that is used to keep the size of the log data to a minimum. If you disable OrientDB Studio, this maintenance process is also disabled, resulting in unrestricted disk usage growth, which has a significant performance impact on the Identity Connect UI.

If you have disabled OrientDB Studio (and the log purging maintenance process) in error, you will need to reenble the maintenance process manually after you have reenabled OrientDB Studio. For more information, see Section 9.2.1, "Purging Synchronization Records".

13.1.2. Changing the OrientDB Admin Password

When you are logged into the repository as the admin user, you have unlimited access to all tables and functions. It is therefore recommended that you change the admin user password in a production system.

To change the OrientDB admin password, follow these steps.

1. Log into the Identity Connect administrative interface, and click Settings.
2. On the settings page, select the Database tab.
3. Type the new password in the OrientDB Admin Password field, and click Save Database Configuration.
4. Restart Identity Connect, as described in Section 2.2, "Stopping and Restarting Identity Connect".

You can also change the admin user password in the OrientDB console, or in OrientDB Studio, as shown in the following examples.

1. To change the admin password in the OrientDB console, make sure that Identity Connect is running, then follow these steps.
 - a. Launch OrientDB console and connect to the database, as described in the previous section.
 - b. Run the following query:

```
orientdb> update ouser set password='password123' where name='admin'  
Updated 1 record(s) in 0.002000 sec(s).
```

This query changes the admin password to `password123`.

To change the admin password in OrientDB Studio, make sure that Identity Connect is running, then follow these steps.

- a. Open OrientDB Studio, as described in the previous section.
 - b. Select Query and enter the following query:

```
update ouser set password='password123' where name='admin'
```
- This query changes the admin password to `password123`.
- c. Click Execute.
2. After you have changed the admin password, shut down Identity Connect, as described in Section 2.2, "Stopping and Restarting Identity Connect".

3. In a text editor, edit the `repo.orientdb.json` file to add a `password` property, with the new value of the password.

```
$ more repo.orientdb.json
{
  "dbUrl" : "&{openidm.repo.orientdb.dburl}",
  "user" : "admin",
  "password" : "password123"
,
...}
```

4. Restart Identity Connect, as described in Section 2.2, "Stopping and Restarting Identity Connect".
5. Check that you can access the OrientDB database with the new password, or monitor the log files to ensure that Identity Connect is able to access the database.

13.1.3. Enabling Database Backups

To enable regular backups of the OrientDB database, follow these steps:

1. Log into the Identity Connect administrative interface, and click Settings.
2. On the settings page, select the Database tab.
3. Check the Enable Database Backups box.
4. In the Backup Directory field, enter the absolute path to the directory in which database backups should be saved. By default, backups are saved to `/path/to/salesforceIdConnect/backups`
5. In the Backup Filename field, specify the name for each new backup file. The default filename is `#{DBNAME}-#{DATE:yyyyMMddHHmmss}.zip`, which amalgamates the database name and the date and time at which the backup was made.

If you enter a static filename here, each successive backup will overwrite the previous backup file.

6. In the First Backup Time field, enter the time at which the initial backup will be made.

The first backup is always made at the specified time, on the day on which the backup configuration is set.

7. In the Backup Interval field, specify the interval between backups. The interval can be expressed with the following units:
 - ms for milliseconds, for example, 10000ms means 10 seconds
 - s for seconds, for example, 10s means 10 seconds
 - m for minutes, for example, 5m means 5 minutes
 - h for hours, for example, 24h means every day

- d for days, for example, 1d means every day

The screenshot shows the 'Database' configuration page in Identity Connect. The page has a navigation bar with tabs: 'Customize Theme', 'SSL Configuration', 'Authentication and Session', 'Database' (selected), and 'About Identity Connect'. Below the navigation bar, there are several configuration sections:

- OrientDB Admin Password:** A text input field.
- Enable OrientDB Studio:** A checkbox that is checked.
- Enable Database Backups:** A checkbox that is checked.
- Backup Configuration Details:**
 - Backup Directory:** A text input field containing '/backups/openidm'.
 - Backup Filename:** A text input field containing '\${DBNAME}-S{DATE:yyyyMMddHHmmss}.zip'.
 - First Backup Time:** A text input field containing '23:59:00'.
 - Backup Interval:** A text input field containing '1d'.

Below the configuration fields, there is a warning message: 'Changes to these settings require a restart of the Identity Connect service.' To the right of this message is a blue button labeled 'Save Database Configuration'. Below the warning message, there is another message: 'Over time, the file size for table indexes is known to expand. From time to time it can be beneficial to rebuild these indexes in order to reduce their sizes.' To the right of this message is a blue button labeled 'Rebuild Audit Indexes'.

8. Click Save Database Configuration.
9. Restart Identity Connect, as described in Section 2.2, "Stopping and Restarting Identity Connect".

13.1.4. Tuning the Performance of the OrientDB Repository

By default, the embedded OrientDB repository assumes an environment with unreliable (non-RAID) hardware. These settings might not be appropriate in other environments.

To improve performance in a deployment that runs on reliable (RAID) hardware, change the following settings in the OrientDB configuration file (`/path/to/salesforceIdConnect/conf/repo.orientdb.json`):

```
"transactionCommitSynch" : false,  
"transactionLogSynch" : false,  
"nonTransactionRecordUpdateSynch" : false
```

By default, these parameters are all set to `true`, which implies the following:

`transactionCommitSynch` - The storage is synchronized after each transaction commit.

`transactionLogSynch` - A disk synch is executed for each log entry, which slows down transactions but guarantees transaction reliability on non-reliable drives.

`nonTransactionRecordUpdateSynch` - A disk synch is executed at every record operation. This slows down record updates but guarantee reliability on unreliable drives.

13.2. Working With Identity Connect Log Files

When you set up Identity Connect by using the `setup.sh` script (on UNIX systems), any startup messages that would be output to the OSGi console are output to the file `/path/to/salesforceIdConnect/logs/console.out`. If you encounter problems while you are configuring Identity Connect, check this file for an indication of what might have gone wrong in the setup process.

On Windows systems, startup messages are output to the Felix shell in the command window in which you launched Identity Connect.

During configuration and authentication, Identity Connect log messages are output to files named `/path/to/salesforceIdConnect/logs/openidm0.log.0`, with the integers being incremented with each successive Identity Connect startup, and after log rotation, when the file size exceeds the configured limit. Check these log files for additional information if you are experiencing problems with Identity Connect.

Log levels and maximum log file sizes are defined in the file `/path/to/salesforceIdConnect/conf/logging.properties`. You can adjust the log level in order to provide more or less information. The default configuration rotates log files when the size reaches 5 MB, and retains up to 5 files.

You can adjust the general log level by changing `.level=INFO` to one of the following, in the `logging.properties` file.

```
SEVERE (highest value)  
WARNING  
INFO  
CONFIG  
FINE  
FINER
```

You can also set specific log levels for individual components. For example, the following setting will provide the maximum output for log messages from the reconciliation process:

```
org.forgerock.openidm.recon.level = FINEST
```

13.3. Using Identity Connect for Delegated Authentication

Identity Connect includes a servlet filter that allows requests from salesforce.com (and any subdomain of salesforce.com) to make AJAX requests to Identity Connect. No specific configuration is required to use this filter. The main purpose of the filter is to provide *delegated authentication*, which enables you to present a standard login form for a specific customer domain (such as `example.salesforce.com`). Instead of submitting login credentials to the Salesforce authentication provider, the filter captures these details and makes a request back to Identity Connect, to obtain the SAML assertion. The SAML assertion can then be submitted to the Salesforce authentication provider, and access is allowed based on that evaluation. Such requests to Identity Connect are transparent - end users do not see the fact that they are actually communicating with a service on premise, rather than with Salesforce itself.

13.4. Synchronizing Passwords With the Active Directory Password Sync Plugin (Advanced Feature)

Caution

Although the password synchronization plugin is a useful tool, it is not the easiest mechanism to achieve common credentials. Solutions such as Delegated Authentication or Federation are generally a better approach for achieving common passwords across your resources. Please consult directly with Salesforce before implementing this feature. Additionally, please ensure that you have Active Directory and certificate management expertise internally, or engage with an implementation partner when implementing this feature.

Password synchronization intercepts user password changes in Active Directory and uses these changes to update the corresponding account in Salesforce. When password synchronization is set up, users authenticate using the same password in both Active Directory and Salesforce. This enables direct authentication with Salesforce, without having to be redirected to Identity Connect. Direct authentication might be necessary when Identity Connect is behind a firewall or is otherwise inaccessible to a user.

Identity Connect can intercept and synchronize passwords that are changed natively in Active Directory and propagate these password changes to Salesforce. To accomplish this synchronization, a filter is deployed on your Active Directory domain controller. The filter captures password changes when they are available in clear text, encrypts them, and passes them to Identity Connect. Identity Connect in turn passes the change to Salesforce and a trigger is used to set the password in Salesforce. If Identity Connect is unavailable when a password change occurs, the password change is queued for subsequent retry. Note that the passwords themselves are never stored in Identity Connect.

If you use password synchronization, you must set up password policy enforcement on Active Directory and ensure that all password policies that are enforced are identical to prevent password updates on one resource from being rejected by Salesforce.

The following sections walk you through the steps required to install the password synchronization plugin and to enable password synchronization between Active Directory and Salesforce. These steps assume that you are running at least Microsoft Windows Server 2008 R2.

Setting up password synchronization involves the following steps:

- Section 13.4.1, "Setting up a Custom Field and Trigger for Password Synchronization"
- Section 13.4.2, "Exporting the Encryption Key"
- Section 13.4.3, "Installing the Password Sync Plugin"

13.4.1. Setting up a Custom Field and Trigger for Password Synchronization

Password synchronization requires that you create a custom user field and a trigger for each Salesforce organization for which you want passwords to be synchronized.

Create a custom password synchronization field as follows:

1. Log in to your Salesforce organization.
2. Click *Setup* in the top right corner.
3. In the left hand menu, under *Build*, select *Customize > Users > Fields*.
4. On the User Fields page, scroll down to the User Custom Fields item and click *New*.
5. On the Data Type page, select *Text* and click *Next*.
6. On the Enter the Details page, provide the following information:
 - **Field Label.** Enter the name of the new field, *PWSync*.
 - **Field Name.** Defaults to the value you set for the *Field Label* (*PWSync*).
 - **Length.** Set the maximum length of the password field to *100*.

You can leave the remaining fields blank.

Step 2. Enter the details **Step 2 of 4**

[Previous](#) [Next](#) [Cancel](#)

Field Label [i](#)

Please enter the maximum length for a text field below.

Length

Field Name [i](#)

Description

Help Text [i](#)

Required Always require a value in this field in order to save a record

Unique Do not allow duplicate values

Treat "ABC" and "abc" as duplicate values (case insensitive)

Treat "ABC" and "abc" as different values (case sensitive)

External ID Set this field as the unique record identifier from an external system

Default Value [Show Formula Editor](#)

Use formula syntax: e.g., Text in double quotes: "hello", Number: 25, Percent as decimal: 0.10, Date expression: Today() + 7

[Previous](#) [Next](#) [Cancel](#)

Click Next to continue.

7. On the Field-Level Security page, limit the field's visibility to System Administrators.

Field-Level Security for Profile	<input type="checkbox"/> Visible	<input type="checkbox"/> Read-Only
Chatter Only User	<input type="checkbox"/>	<input type="checkbox"/>
Company Communities User	<input type="checkbox"/>	<input type="checkbox"/>
Contract Manager	<input type="checkbox"/>	<input type="checkbox"/>
Customer Community User	<input type="checkbox"/>	<input type="checkbox"/>
Force.com - App Subscription User	<input type="checkbox"/>	<input type="checkbox"/>
Identity User	<input type="checkbox"/>	<input type="checkbox"/>
Marketing User	<input type="checkbox"/>	<input type="checkbox"/>
Read Only	<input type="checkbox"/>	<input type="checkbox"/>
Solution Manager	<input type="checkbox"/>	<input type="checkbox"/>
Standard User	<input type="checkbox"/>	<input type="checkbox"/>
System Administrator	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Work.com Only User	<input type="checkbox"/>	<input type="checkbox"/>

Click Next.

- On the Add to Page Layouts page, deselect *Add Field*.

Click Save.

Create an Apex trigger as follows:

- On your organization's Setup page, in the left hand menu, under *Build*, select *Customize > Users > Triggers*.
- On the User Triggers page, click *New*.
- On the Apex Trigger tab, paste the following script.

```
trigger PWSync on User (before insert, before update) {
    for(User u: Trigger.new){
        if (u.PWSync__c != null ) {
            System.setPassword(u.Id, u.PWSync__c);
            u.PWSync__c = null;
        }
    }
}
```

The Apex Trigger tab should now look as follows:

Apex Trigger

Apex Trigger Edit

Save Quick Save Cancel

```

1 trigger PWSync on User (before insert, before update) {
2
3     for(User u: Trigger.new){
4         if (u.PWSync__c != null ) {
5             System.setPassword(u.Id, u.PWSync__c);
6             u.PWSync__c = null;
7         }
8     }
9
10 }

```

4. Click Save to save the new trigger.

13.4.2. Exporting the Encryption Key

The password synchronization plugin encrypts passwords using the Advanced Encryption Standard (AES). The AES key is encrypted using Identity Connect's public key. Identity Connect then uses its private key to decrypt the AES key and then uses the AES key to decrypt the password.

This section describes how to export Identity Connect's public key and certificate so that it can be used by the password synchronization plugin to encrypt the AES key. The same certificate is needed by the password synchronization plugin on the Active Directory host to trust the SSL certificate provided by Identity Connect over REST.

The certificate that Active Directory uses to authenticate to OpenIDM must be configured with an appropriate encoding, cryptographic hash function, and digital signature. The plugin can read a public or a private key from a PKCS12 archive file. For production purposes, you should use a certificate that has been issued by a Certificate Authority. For testing purposes, you can use the self-signed certificate that is generated by Identity Connect. Whichever certificate you use, you must import that certificate into OpenIDM's trust store, as shown in the following procedure.

The plugin itself will be installed on your Active Directory Domain Controller in the next section.

Important

Encryption of the password over the network relies on a secure (SSL) connection between Identity Connect and the Active Directory host - that is, a connection over https, using a secure port. If the connection between Identity Connect and Active Directory is over plain http, the password is sent in clear text.

By default, the plugin does not validate the Identity Connect certificate. In a production environment, you should configure certificate validation by setting the following registry key: `netSslVerifyPeer = True`. For more information, see Section 13.4.5, "Changing the Password Synchronization Plugin Configuration After Installation".

1. Export the certificate for the `openidm-localhost` entry.

Use the `-rfc` option to print the certificate in PEM format.

The default keystore password is `changeit`.

```
$ cd /path/to/salesforceIdConnect/security
$ keytool \
  -export \
  -alias openidm-localhost \
  -file openidm-localhost-cert.crt \
  -keystore keystore.jceks \
  -storetype jceks \
  -rfc \
  -storepass changeit

Certificate stored in file <openidm-localhost-cert.crt>
```

2. Export the public key and certificate as a `.p12` file, named `openidm-localhost.p12`.

The `Export Password` that you enter here will be used to open the file. You will need this export password when you set up the password synchronization plugin, in the following section. This example uses `Passw0rd` for the export password.

```
$ openssl pkcs12 \
  -export \
  -nokeys \
  -in openidm-localhost-cert.crt \
  -out openidm-localhost.p12

Enter Export Password: <Passw0rd>
Verifying - Enter Export Password: <Passw0rd>
```

3. Copy the `p12` certificate file (`openidm-localhost.p12`) that you created in the previous step to your Active Directory Domain Controller.

The following procedure assumes that you have copied the file to `C:/Users/Administrator/openidm-localhost.p12`.

Important

There is currently an issue relating to a mismatch between the certificate that is generated when Identity Connect starts up, and the default certificate that is in Identity Connect's truststore. The issue can cause the password synchronization plugin to fail. To work around the issue, you must export the certificate from the keystore to the truststore, as follows:

1. Remove the default certificate from Identity Connect's truststore.

```
$ cd /path/to/salesforceIdConnect/security
$ keytool \
  -delete \
  -alias openidm-localhost \
  -keystore truststore \
  -storetype JKS \
  -storepass changeit
```

2. Import the certificate that you exported in Step 1 of the previous procedure (`openidm-localhost-cert.crt`) into the truststore.

```
$ keytool \
  -import \
  -file openidm-localhost-cert.crt \
  -alias openidm-localhost \
  -keystore truststore \
  -storetype JKS \
  -storepass changeit
Owner: C=None, L=None, O=OpenIDM Self-Signed Certificate, OU=None, CN=localhost
Issuer: C=None, L=None, O=OpenIDM Self-Signed Certificate, OU=None, CN=localhost
Serial number: 3d9fd5e0e7fabe9a
Valid from: Sat Jul 25 21:06:56 SAST 2015 until: Thu Aug 21 21:06:56 SAST 2025
Certificate fingerprints:
  MD5:  8F:AB:00:71:E1:D7:B6:84:E6:55:F3:B1:CC:86:8B:9D
  SHA1: 42:8E:EE:6F:5A:E1:64:F6:4C:CC:51:BC:B2:01:C8:77:69:00:04:A4
  SHA256: 5B:78:D5:0F:92:87:D4:FC:AF:C9:C6:53:03:C7:5B:2A:0B:...E5:CC:DF:97:1C:9A:16
Signature algorithm name: SHA512withRSA
Version: 3
Trust this certificate? [no]: yes
Certificate was added to keystore
```

3. Restart Identity Connect.

```
$ cd /path/to/salesforceIdConnect
$ $ ./shutdown.sh
./shutdown.sh
Stopping OpenIDM (9184)
$ ./startup.sh
Executing ./startup.sh...
Using OPENIDM_HOME: /path/to/salesforceIdConnect
Using OPENIDM_OPTS: -Xmx1024m -Xms1024m -Dstorage.wal.maxSize=500
-Dlogback.configurationFile=conf/logging-config.xml
Using LOGGING_CONFIG:
-Djava.util.logging.config.file=/path/to/salesforceIdConnect/conf/logging.properties
Using boot properties at /path/to/salesforceIdConnect/conf/boot/boot
.properties
-> OpenIDM ready
```

13.4.3. Installing the Password Sync Plugin

The password sync plugin is provided with the Identity Connect delivery and must be installed on each Active Directory Primary Domain Controller. The plugin intercepts password changes and

sends updated password values to Identity Connect over an encrypted channel. You must have Administrator privileges to install the plugin.

1. After you have extracted the Identity Connect installation, change to the `bin` directory.

```
C:\>cd salesforceIdConnect\bin
```

2. Copy the plugin setup file (`ad-passwordchange-handler.exe`) to a location on your Active Directory Domain Controller.

3. Launch the password sync setup wizard.

```
C:\path\to>ad-passwordchange-handler.exe
```



4. Accept the Common Development and Distribution License (CDDL) agreement to proceed with the installation.
5. On the OpenIDM Information Connection screen, provide the following information:

- **OpenIDM URL.** Enter the URL at which Identity Connect is accessed (including the port) plus the following endpoint and query `/openidm/endpoint/sfpwdplugin?_action=patch-by-query&_uid=${samaccountname}`

For example:

```
https://connect.example.com:8443/openidm/endpoint/sfpwdplugin?_action=patch-by-query&_uid=${samaccountname}
```

Note that the Active Directory server must be able to access this URL directly.

- **OpenIDM User Password attribute.** The Identity Connect implementation of the password sync plugin ignores this value, so you can leave the default (`adPassword`).

Setup - OpenIDM Password Sync

OpenIDM Information
Connection

Please specify OpenIDM server deployment URL and request template information, then click Next.

OpenIDM URL:

OpenIDM User Password attribute:

< Back Next > Cancel

6. On the Authentication screen, enter the credentials of a user that has administrative privileges in Identity Connect.

Select *OpenIDM Header* as the authentication type.

Setup - OpenIDM Password Sync

OpenIDM Information
Authentication

Please specify OpenIDM authentication parameters, then click Next.

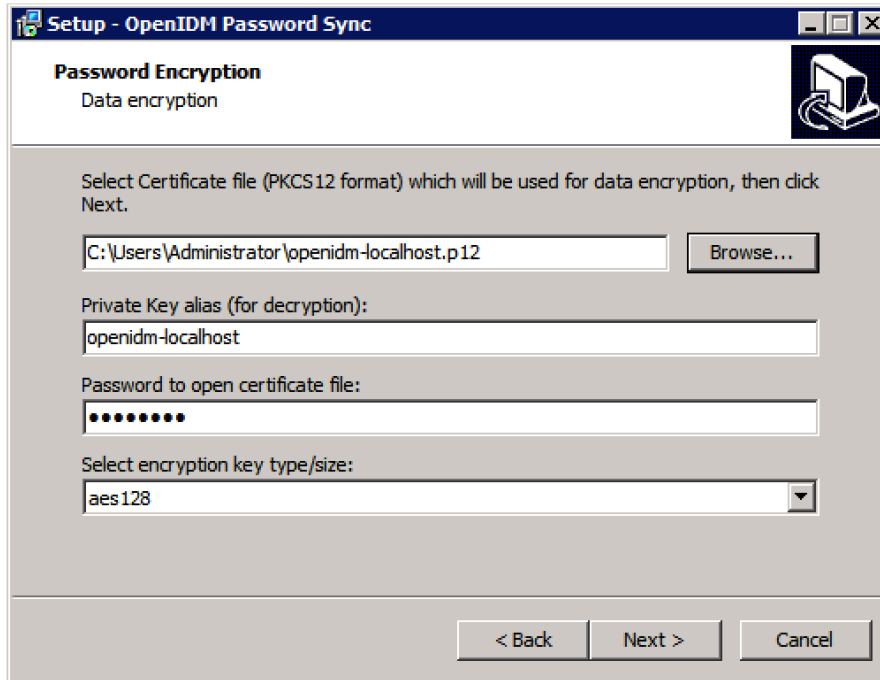
User name:
administrator

Password:
.....

Select authentication type:
OpenIDM Header

< Back Next > Cancel

7. On the Password Encryption screen, provide the following information:
 - **Certificate file.** Browse to locate the p12 certificate file that you copied previously.
 - **Private key alias.** Specify the name of the p12 certificate file (`openidm-localhost` in our example).
 - **Password to open certificate file.** Enter the export password that you chose when you created the p12 certificate file.
 - **Select encryption key type.** Specify the encryption key type that will be used when encrypting the password value (AES-128, AES-192, or AES-256).



If you select an encryption key type greater than AES-128, you must install the Unlimited JCE Policy for your JRE, *on the machine on which Identity Connect is installed*. To install the unlimited JCE Policy, follow these steps:

- Download the JCE zip file for Java 8 from the Oracle Technetwork site.
- Locate the `lib\security` folder of your JRE, for example, `C:\Program Files\Java\jre8\lib\security`.
- Remove the following `.jar` files from the `lib\security` folder:

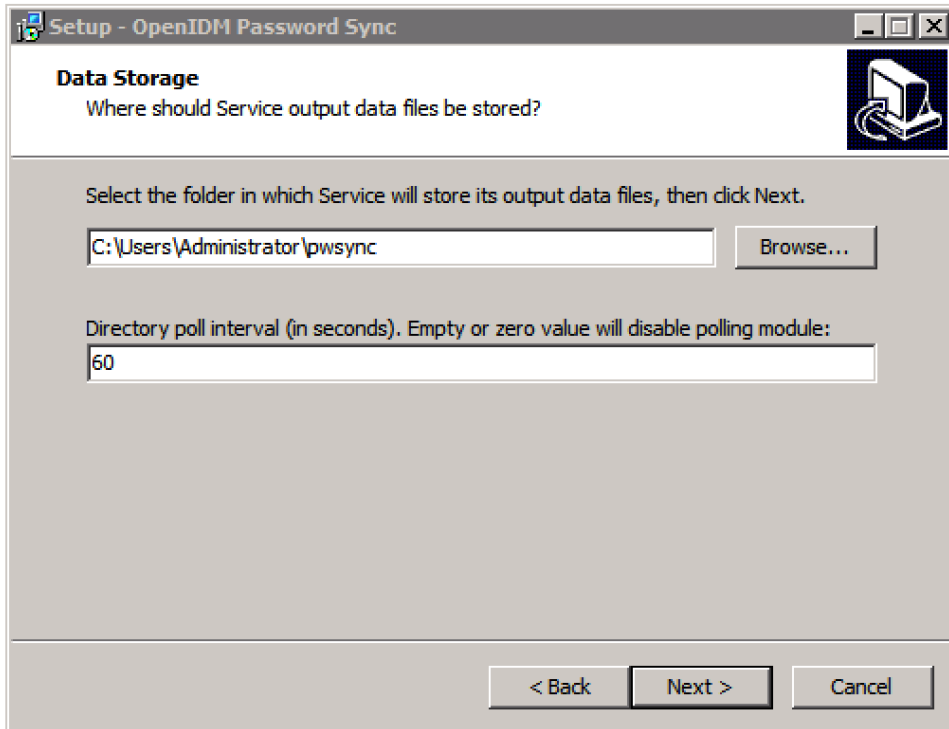
```
local_policy.jar
US_export_policy.jar
```

- Unzip the JCE zip file and copy the two `_policy.jar` files to the `lib\security` folder of your JRE.
 - If Identity Connect is already running, you must restart it for the installation of the JCE policy files to take effect.
8. On the Data Storage screen, provide the following information:
- Browse for the folder in which queued password changes will be stored.

If Identity Connect cannot be reached when a password change is made on Active Directory, the change is placed in this queue. All outstanding password changes are propagated when Identity Connect becomes available.

Because this folder contains password information, it is strongly recommended that you restrict access to this folder to administrators.

- Specify the interval at which the data storage queue should be polled for changes. The default interval is 60 seconds.

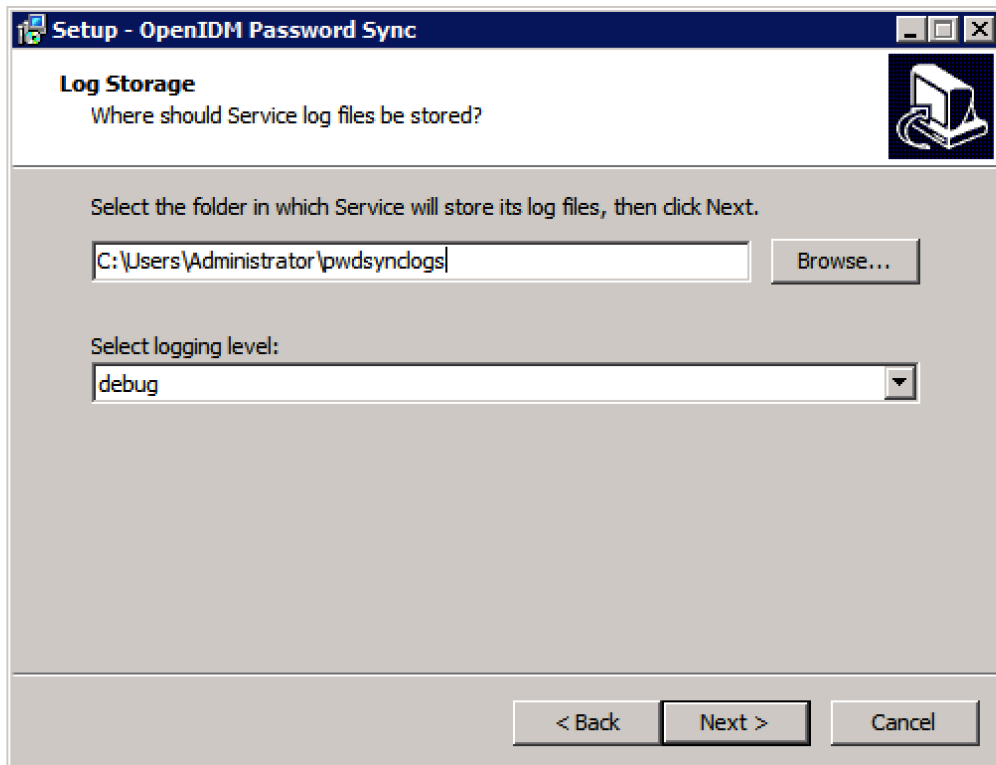


9. On the Log Storage screen, provide the following information:

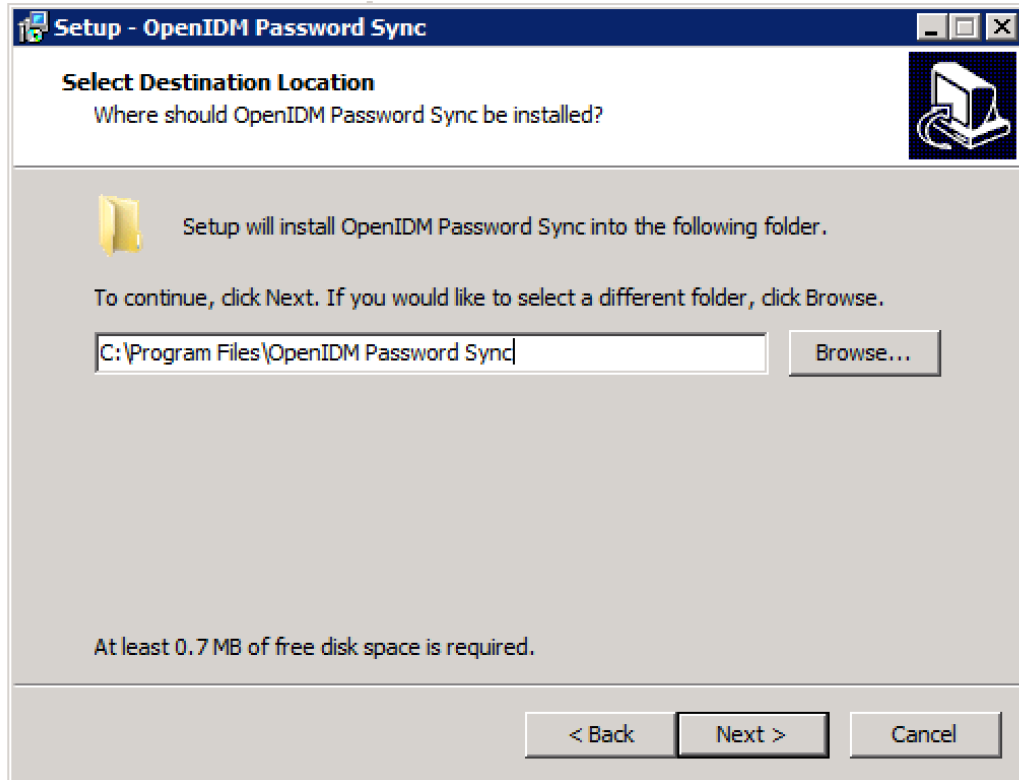
- Browse for the folder in which log files will be stored.
- Specify the level at which messages will be logged. The amount of information that is logged corresponds to the following log levels, from the logging all messages (**debug**) to logging only fatal errors (**fatal**).
 - **debug**
 - **info**

- warning
- error
- fatal

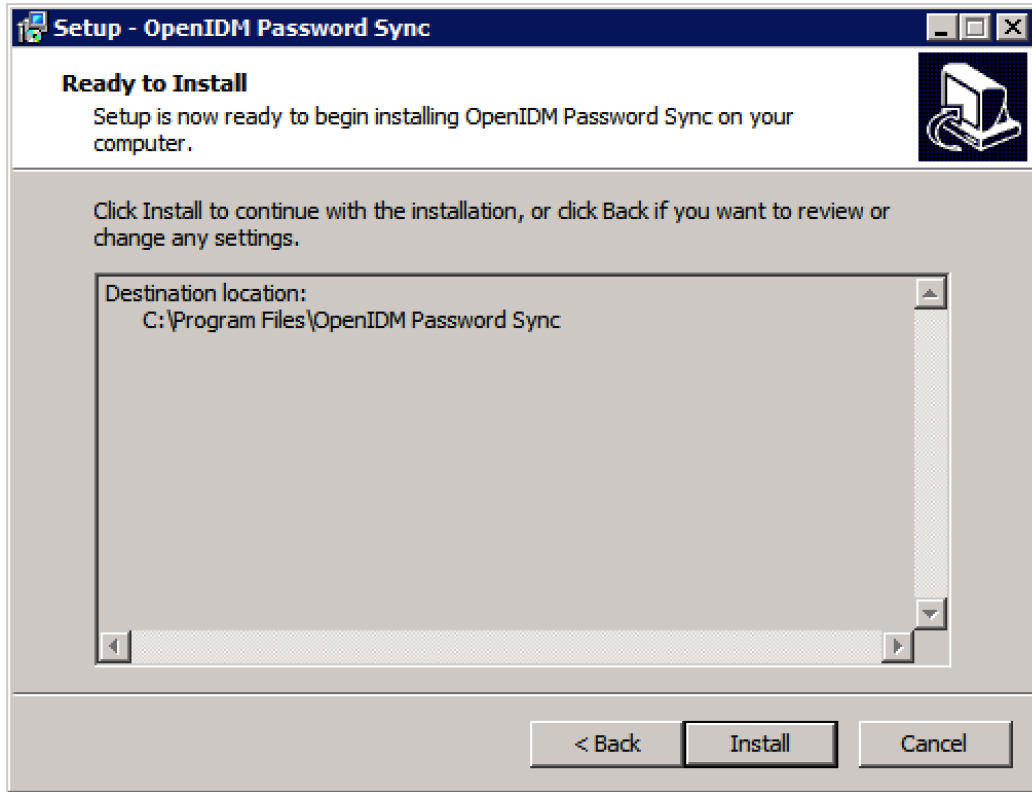
In general, you should set the log level to `debug` or `info` in production, to ensure that you capture enough information to help diagnose issues.



10. Select the directory in which the password synchronization plugin will be installed.



11. Click Install to complete the installation process.



12. When the installation is complete, you must restart your Active Directory Domain Controller for the password synchronization plugin to start working.

Click Finish to restart the Domain Controller.



13.4.4. Testing Password Synchronization

After you have installed the password synchronization plugin, test that it is working by changing the password of an Active Directory user. The password change can take some time to be propagated to Salesforce, after which you should be able to log into Salesforce with the new credentials.

If the synchronization is not successful, you should see the change request in the password queue location that you specified when you installed the plugin. You can remove the change request by deleting the file.

The debug logs are located in the password plugin log directory that you specified when you installed the plugin. Additional debugging information is available in the `/path/to/salesforceIdConnect/logs/openidm.log` file.

13.4.5. Changing the Password Synchronization Plugin Configuration After Installation

If you need to change any settings after installation, access the settings using the Registry Editor under HKEY_LOCAL_MACHINE > SOFTWARE > ForgeRock > OpenIDM > PasswordSync.

For information about creating registry keys, see [Configure a Registry Item in the Windows documentation](#).

You can change the following registry keys to reconfigure the plugin:

Keys to set the method of authentication

- `authType` sets the authentication type.

For plain HTTP or SSL authentication using OpenIDM headers, set `authType` to `idm`.

For SSL mutual authentication using a certificate, set `authType` to `cert`.

By default, the plugin does not validate the Identity Connect certificate. To configure this validation, set the following registry key: `netSslVerifyPeer = True`.

- `authToken0` sets the username or certificate path for authentication.

For example, for plain HTTP or SSL authentication, set `authToken0` to `admin`.

For SSL mutual authentication, set `authToken0` to the certificate path, for example `path/to/certificate/cert.p12`. Only PKCS12 format certificates are supported.

- `authToken1` sets the password for authentication.

For example, for plain HTTP or SSL authentication, set `authToken1` to `admin`.

For SSL mutual authentication, set `authToken1` to the password to the keystore.

Keys to set encryption of captured passwords

- `certFile` sets the path to the keystore used for encrypting captured passwords, for example `path/to/keystore.p12`. Only PKCS12 keystores are supported.
- `certPassword` sets the password to the keystore.
- `keyAlias` specifies the alias that is used to encrypt passwords.
- `keyType` sets the cypher algorithm, for example `aes128`

Key to set the connection information

Reset the following key to change the connection information that you specified during setup:

- `idmURL` - the URL at which Identity Connect is accessed (including the port) plus the following endpoint and query `/openidm/endpoint/sfpwdplugin?_action=patch-by-query&_uid=${samaccountname}`

Keys to set the behavior when Identity Connect is unavailable

When Identity Connect is unavailable, or when an update fails, the password synchronization plugin stores the user password change a JSON file on the Active Directory system and attempts to resend the password change at regular intervals.

After installation, you can change the behaviour by setting the following registry keys:

- `dataPath` - the location where the plugin stores the unsent changes. When any unsent changes have been delivered successfully, files in this path are deleted. The plugin creates one file for each user. This means that if a user changes his password three times in a row, you will see only one file containing the last change.
- `pollEach` - the interval (in seconds) at which the plugin attempts to resend the changes.

Keys to set the logging configuration

- `logPath` sets the path to the log file.
- `logLevel` sets the logging level, `debug`, `info`, `warning`, `error`, or `fatal`.

13.5. Managing Scheduled Tasks in Identity Connect

Certain Identity Connect tasks, such as the purging of log files, and specific reconciliation operations, are provided with default schedules. This section lists the default scheduled tasks and indicates how you can change the times at which they run.

Scheduled tasks are configured in files in the `/path/to/salesforceIdConnect/conf` directory. The following schedules are provided by default:

- `schedule-autoPurgeAuditRecon_ADUsers_SalesForceUsers.json`

Scheduled operation that purges the audit records used for synchronization reporting (see Section 9.2.1, "Purging Synchronization Records").

- `schedule-autoPurgeAuditRecon_ADUsers_SalesForceUsers_Analysis.json`

Scheduled operation that purges audit records for mappings for which the UI needs only one summary record (see Section 9.2.1, "Purging Synchronization Records").

- `schedule-salesforce_org-id_SFPSA_Managed.json`

Scheduled operation to synchronize Salesforce permission set assignments.

- `schedule-salesforce_ADGroups_ManagedGroups.json`

Scheduled operation to synchronize Active Directory groups with the groups stored in Identity Connect.

- `schedule-livesyncADGroups.json`

Scheduled operation to synchronize Active Directory groups with Salesforce groups.

- `schedule-salesforceDataSyncNow.json`

Scheduled operation to synchronize Active Directory user accounts with Salesforce users.

The scheduled data synchronization that you configure on the Sync tab should be adjusted by using the UI, rather than by manipulating the corresponding schedule files (`schedule-livesyncADGroups.json` and `schedule-salesforceDataSyncNow.json`) directly. For more information, see Section 5.3, "Configuring the Synchronization Schedule".

For other scheduled operations, you can change the schedules by modifying the corresponding schedule configuration file. All schedule configuration files have the following format:

```
{
  "enabled"           : true,
  "persisted"        : true,
  "concurrentExecution" : false,
  "type"              : "cron",
  "schedule"          : "quartz expression",
  "misfirePolicy"     : "optional, string",
  "invokeService"     : "service identifier",
  "invokeContext"     : "service specific context info",
  "invokeLogLevel"   : "optional, level"
}
```

The schedule configuration properties are as follows:

enabled

Set to `true` to enable the schedule. When this property is set to `false`, Identity Connect considers the schedule configuration dormant, and does not allow it to be triggered or executed.

If you want to retain a schedule configuration, but do not want it used, set `enabled` to `false`, instead of changing the configuration or `quartz` expression.

persisted (optional)

Specifies whether the schedule state should be persisted or stored in RAM. Boolean (`true` or `false`), `false` by default.

In a clustered environment, this property must always be set to `true` to have the schedule fire only once across the cluster.

concurrentExecution

Specifies whether multiple instances of the same schedule can run concurrently. Boolean (`true` or `false`), `false` by default. Multiple instances of the same schedule cannot run concurrently by default. This setting prevents a new scheduled task from being launched before the same

previously launched task has completed. For example, under normal circumstances you would want a live update operation to complete its execution before the same operation was launched again. To enable concurrent execution of multiple schedules, set this parameter to `true`.

type

Identity Connect supports only `cron` schedules.

schedule

Takes **quartz** expression syntax. For more information about **quartz** syntax, see the *CronTrigger Tutorial*.

misfirePolicy

For persistent schedules, this optional parameter specifies the behavior if the scheduled task is missed, for some reason. Possible values are as follows:

- `fireAndProceed`. The first execution of a missed schedule is immediately executed when Identity Connect is back online. Subsequent executions are discarded. After this, the normal schedule is resumed.
- `doNothing`, all missed schedules are discarded and the normal schedule is resumed when Identity Connect is back online.

invokeService

Defines the type of scheduled event or action. The value of this parameter can be one of the following:

- `sync` for reconciliation
- `provisioner` for LiveSync
- `script` to call some other scheduled operation defined in a script

invokeContext

Specifies contextual information, depending on the type of scheduled event (the value of the `invokeService` parameter).

For example, following excerpt of a schedule configuration invokes the script that purges the reconciliation audit logs.

```
...
  "invokeService" : "script",
  "invokeContext" : {
    "script" : {
      "type" : "text/javascript",
      "file" : "script/autoPurgeAuditRecon.js"
    },
  },
  ...
```

Chapter 14

Troubleshooting an Identity Connect Installation

This chapter describes common problems that might occur during the installation and configuration of Identity Connect, and how these problems can be resolved.

14.1. Troubleshooting the Integrated Windows Authentication Configuration

This section describes problems that might occur during the configuration and use of Integrated Windows Authentication (IWA) with Identity Connect. The IWA configuration process is described in Chapter 7, "*Configuring Identity Connect for Integrated Windows Authentication (Advanced Feature)*".

The IWA setup involves three broad steps:

1. Configuring a Kerberos user account and creating a keytab file
2. Configuring the authentication filter in Identity Connect
3. Configuring the client browser to support SPNEGO

This troubleshooting section is broken down into those steps, to enable you to pinpoint problems in the configuration.

14.1.1. Configuring the Kerberos User account and Creating the Keytab File

The creation of a keytab and the configuration of a dedicated Active Directory user for the service are critical elements of a Kerberos configuration. Before describing potential problems with these elements, it is helpful to have an understanding of the main Kerberos components involved in the authentication process.

- Kerberos distinguishes between two types of principals (accounts) - User Principal Name (UPN), and Service Principal Name (SPN). Both of these are essentially unique identifiers for the security identity of a user or of a computer. UPNs are of the format `userID@DNSDomainName` while SPNs are of the format `serviceClass/host:port/serviceName`.

Both UPNs and SPNs are registered in the Active Directory Domain Controller (DC) for the user account that the Identity Connect instance will use.

Kerberos authentication uses SPNs to identify the specific services to which clients have access. The first time a client requests authentication, the client must include the SPN of the Identity Connect service in its request. To do so, the Kerberos user account must be linked to the SPN of the service.

- The Key Distribution Center (KDC) comprises two elements - the Authentication Service and the Ticket Granting Service. Identity Connect uses its keytab to authenticate against the Authentication Service (AS) and obtains a ticket from the Ticket Granting Service (TGS) for the specified Service Principal Name (SPN).
- The AS uses the SPN to locate the service user entry in Active Directory and to retrieve the account password to establish a session key.

The following scenarios and misconfigurations can cause errors at this point.

Incorrect UPN

If the user account uses a UPN that does not match the SPN that Identity Connect uses (and the SPN that is defined in the keytab), an error similar to the following is output:

```
====
Sep 6, 2013 4:33:52 AM org.forgerock.jaspi.modules.iwa.wdssso.WDSSO serviceLogin
SEVERE: IWA WDSSO: Service Login Error: Client not found in Kerberos database (6)
Sep 6, 2013 4:33:52 AM org.forgerock.jaspi.modules.iwa.wdssso.WDSSO serviceLogin
SEVERE: IWA WDSSO: Stack trace:
javax.security.auth.login.LoginException: Client not found in Kerberos database
(6) at com.sun.security.auth.module.Krb5LoginModule.attemptAuthentication
(Krb5LoginModule.java:759)
at com.sun.security.auth.module.Krb5LoginModule.login(Krb5LoginModule.java:580)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke
(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at javax.security.auth.login.LoginContext.invoke(LoginContext.java:784)
at javax.security.auth.login.LoginContext.access$000(LoginContext.java:203)
at javax.security.auth.login.LoginContext$4.run(LoginContext.java:698)
at javax.security.auth.login.LoginContext$4.run(LoginContext.java:696)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.login.LoginContext.invokePriv(LoginContext.java:695)
at javax.security.auth.login.LoginContext.login(LoginContext.java:594)
at org.forgerock.jaspi.modules.iwa.wdssso.WDSSO.serviceLogin(WDSSO.java:601)
...
Caused by: KrbException: Client not found in Kerberos database (6)
at sun.security.krb5.KrbAsRep.<init>(KrbAsRep.java:76)
at sun.security.krb5.KrbAsReqBuilder.send(KrbAsReqBuilder.java:319)
at sun.security.krb5.KrbAsReqBuilder.action(KrbAsReqBuilder.java:364)
at com.sun.security.auth.module.Krb5LoginModule.attemptAuthentication
(Krb5LoginModule.java:731)
... 61 more
Caused by: KrbException: Identifier doesn't match expected value (906)
at sun.security.krb5.internal.KDCRep.init(KDCRep.java:143)
at sun.security.krb5.internal.ASRep.init(ASRep.java:65)
at sun.security.krb5.internal.ASRep.<init>(ASRep.java:60)
```



```
at sun.security.krb5.KrbAsRep.<init>(KrbAsRep.java:60)
... 64 more
===
```

The message does, in fact, identify the issue, which occurs during the authentication attempt. In this case, the SPN name that is used by Identity Connect was not found in the Kerberos database (Active Directory).

The UPN, or user logon name, is the crucial attribute in this error. The UPN *must* match the SPN that Identity Connect uses.

After you have run the **ktpass** to create the keytab file, you can query the Kerberos user account to check the UPN and SPN that were added to the account. You can use the freely available **ldapsearch** command-line utility, or any other LDAP browser.

The following command uses **ldapsearch** to query the Kerberos user account.

```
$ ./ldapsearch \
--hostname ad-host \
--port 389 \
--bindDN "cn=administrator,cn=users,dc=ad,dc=example,dc=com" \
--bindPassword Secret12! \
--bindDN "cn=users,dc=ad,dc=example,dc=com" \
"(cn="Identity Connect")"
dn: CN=Identity Connect,CN=Users,DC=ad,DC=example,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Identity Connect
sn: Connect
givenName: Identity
distinguishedName: CN=Identity Connect,CN=Users,DC=ad,DC=example,DC=com
instanceType: 4
...
sAMAccountName: identityconnect
sAMAccountType: 805306368
userPrincipalName: HTTP/connect.ad.example.com@AD.EXAMPLE.COM
servicePrincipalName: HTTP/connect.ad.example.com
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=ad,DC=example,DC=com
```

Note the values of the **userPrincipalName** and **servicePrincipalName** to this user account.

Important

This error might *also* be output if you have associated multiple Active Directory accounts with a single SPN. For more information, see [Single SPN associated with multiple accounts](#).

Inconsistent Key Version Number (kvno)

When you create a keytab without specifying a key version number (using the **ktpass** command without the **kvno** option), the **msDS-KeyVersionNumber** is automatically incremented in Active Directory. You can obtain the current key version number by using the **klist** command, for example:

```
$ klist -ke -t security/identityConnect.HTTP.keytab
Keytab name: FILE:security/identityConnect.HTTP.keytab
KVNO Timestamp          Principal
-----
  5 01/01/70 00:00:00 HTTP/connect.forgerock.com@ad.example.com
```

If the key version number is incorrect, an error similar to the following is observed:

```
====
Sep 26, 2013 6:03:49 AM org.forgerock.jaspi.modules.iwa.wdsso.WDSSO process
SEVERE: IWA WDSSO: Authentication failed with GSSEException. Failure unspecified
at GSS-API level
(Mechanism level: Specified version of key is not available (44))
=====
```

Single SPN associated with multiple accounts

If the same SPN is associated with multiple Active Directory accounts, the Kerberos exchange will fail, because the Key Distribution Center is unable to determine which entry to use.

The error message might be confusing but is generally an indication that multiple accounts have been associated with the SPN:

```
====
Sep 26, 2013 6:21:36 AM org.forgerock.jaspi.modules.iwa.wdsso.WDSSO process
SEVERE: IWA WDSSO: Authentication failed with GSSEException. Defective token
detected (Mechanism level: GSSHeader did not find the right tag)
=====
```

To check whether multiple accounts are linked to the same SPN, search the Active Directory Forest for all accounts linked to that SPN. Execute the following command on the Domain Controller:

```
setspn -Q {SERVICE_PRINCIPAL_NAME}
```

where `{SERVICE_PRINCIPAL_NAME}` is the SPN that you specified when you created the keytab file. This command lists all the Active Directory accounts that are associated with that SPN.

Use of a high strength cipher for the keytab

If you need to use a higher strength cryptosystem, such as `AES256-SHA1`, the following additional configuration is required:

1. Download and install the Unlimited JCE Policy for Java 8 from the [Oracle Technetwork site](#).
2. Unzip the JCE zip file and install the JCE policy jar files in the `/lib/security` folder of the JRE.
3. When you have installed the Unlimited JCE policy, configure the `identityconnect` user entry to support AES 256-bit encryption.

Select the `identityconnect` user entry and select Properties.

On the Account tab, select AES 256 under Account Options.

The screenshot shows the 'Salesforce Bridge Properties' dialog box with the 'Account' tab selected. The 'User logon name' is 'iconnect@ad.example.com' and the 'User logon name (pre-Windows 2000)' is 'AD0\iconnect'. The 'Account options' section is expanded, showing the following options:

- Use Kerberos DES encryption types for this account
- This account supports Kerberos AES 128 bit encryption.
- This account supports Kerberos AES 256 bit encryption.
- Do not require Kerberos preauthentication

The 'Account expires' section is set to 'Never'.

If you do not configure the `identityconnect` user entry to support AES 256-bit encryption, the following error is displayed in the log:

```
Jul 24, 2014 4:26:59 PM org.forgerock.jaspi.modules.iwa.wdsso.WDSSO process
SEVERE: IWA WDSSO: Authentication failed with GSSException. Failure unspecified
at GSS-API level (Mechanism level: Invalid argument (400) - Cannot find key of
appropriate type to decrypt AP REP - RC4 with HMAC)
```

If the JCE Unlimited Policy files are not installed, an error similar to the following is seen in the logs when a `WWW-Authenticate : Negotiate` takes place:

```
Jul 23, 2014 8:34:19 PM org.forgerock.jaspi.modules.iwa.wdsso.WDSSO serviceLogin
SEVERE: IWA WDSSO: Service Login Error: Unable to obtain password from user
```

```
Jul 23, 2014 8:34:19 PM org.forgerock.jaspi.modules.iwa.wdsso.WDSSO serviceLogin
SEVERE: IWA WDSSO: Stack trace:
javax.security.auth.login.LoginException: Unable to obtain password from user

    at com.sun.security.auth.module.Krb5LoginModule.promptForPass(Unknown Source)
    at com.sun.security.auth.module.Krb5LoginModule.attemptAuthentication(Unknown Source)
    at com.sun.security.auth.module.Krb5LoginModule.login(Unknown Source)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
    at java.lang.reflect.Method.invoke(Unknown Source)
    at javax.security.auth.login.LoginContext.invoke(Unknown Source)
    at javax.security.auth.login.LoginContext.access$000(Unknown Source)
    at javax.security.auth.login.LoginContext$4.run(Unknown Source)
    at javax.security.auth.login.LoginContext$4.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.login.LoginContext.invokePriv(Unknown Source)
    at org.forgerock.jaspi.modules.iwa.wdsso.WDSSO.serviceLogin(WDSSO.java:601)
    at org.forgerock.jaspi.modules.iwa.wdsso.WDSSO.initWindowsDesktopSSOAuth(WDSSO.java:560)
    at org.forgerock.jaspi.modules.iwa.wdsso.WDSSO.process(WDSSO.java:139)
    at org.forgerock.jaspi.modules.iwa.IWAModule.validateRequest(IWAModule.java:107)
    at org.forgerock.openidm.jaspi.modules.IWAModule.validateRequest(IWAModule.java:105)
    at org.forgerock.openidm.jaspi.modules.IWAPassthroughModule.validateRequest
      (IWAPassthroughModule.java:114)
    at org.forgerock.openidm.jaspi.modules.IDMServerAuthModule.validateRequest
      (IDMServerAuthModule.java:139)
    at org.forgerock.jaspi.container.ServerAuthContextImpl.validateRequest
      (ServerAuthContextImpl.java:177)
    at org.forgerock.jaspi.filter.AuthNFilter.doFilter(AuthNFilter.java:162)
```

This issue is similar to the issue of the SPN in the Identity Connect configuration not matching what is in the keytab file. Essentially, based on the cipher that is used in the keytab, Identity Connect cannot locate a valid key to use (not because the SPN does not match, but because it cannot locate an SPN with a valid cipher in the keytab).

14.1.2. Configuring the Authentication Filter in Identity Connect

This section highlights common errors that occur while configuring IWA in the Identity Connect admin console. These errors might result in some fairly cryptic messages being output. The errors can usually be resolved by updating the IWA configuration in the Identity Connect Administration interface, or by updating the configuration files in the `path/to/salesforceIdConnect/security` directory.

Client and Identity Connect on the same node

The client (browser) and Identity Connect *must* be running on different hosts if you are using IWA. If they are on the same host, no ticket will be available to the client. In this case, the following error is logged:

```
SEVERE: IWA WDSSO: Authentication failed with GSSEException. Defective token
detected (Mechanism level: GSSHeader did not find the right tag)
```

Incorrect KDC server name specified in Identity Connect

If the name of the Key Distribution Center (KDC) server is incorrect, an error similar to the following is output to the openidm log files:

```
====  
org.forgerock.jaspi.modules.iwa.wdsso.WDSSO serviceLogin  
SEVERE: IWA WDSSO: Service Login Error: server-name: Name or service not known  
org.forgerock.jaspi.modules.iwa.wdsso.WDSSO serviceLogin  
SEVERE: IWA WDSSO: Stack trace:  
javax.security.auth.login.LoginException: server-name: Name or service not  
known at com.sun.security.auth.module.Krb5LoginModule.attemptAuthentication  
(Krb5LoginModule.java:763)  
.....  
Caused by: java.net.UnknownHostException: server-name: Name or service not  
known at java.net.InetAddressImpl.lookupAllHostAddr(Native Method)  
at java.net.InetAddress$1.lookupAllHostAddr(InetAddress.java:894)  
====
```

This error should be resolved when you specify the correct name for the KDC server.

In the Identity Connect administration interface, click Settings and select the Authentication and Session tab. Enter the correct KDC server name in the Windows Domain Controller field.

Incorrect SPN (Service Principal Name)

If the SPN that is specified in the Identity Connect configuration does not match the SPN that is provided in the keytab, Identity Connect is unable to acquire its login information. The following error is output to the openidm log files:

```
====  
org.forgerock.jaspi.modules.iwa.wdsso.WDSSO serviceLogin  
SEVERE: IWA WDSSO: Service Login Error: Unable to obtain password from user  
  
org.forgerock.jaspi.modules.iwa.wdsso.WDSSO serviceLogin  
SEVERE: IWA WDSSO: Stack trace:  
javax.security.auth.login.LoginException: Unable to obtain password from user  
...  
====
```

The message in the stack trace can be confusing. It indicates, however, that during the module initialization, the `promptForPass()` method of the `Krb5LoginModule.java` module fails while attempting to validate the principal (SPN), by using the keytab.

This error should be resolved when you provide an SPN in the Identity Connect configuration that matches the keytab.

In the Identity Connect administration interface, click Settings and select the Authentication and Session tab. Enter the correct SPN name in the Service Principal Name (SPN) field.

Case sensitive realm name in the SPN (Service Principal Name)

The realm name must be written in upper case. The following is an example of a valid SPN name:

```
HTTP/connect.ad.example.com@AD.EXAMPLE.COM
```

If the realm name is in lower case, for example:

```
HTTP/connect.ad.example.com@ad.example.com
```

errors such as the following are output to the logs:

```
====  
SEVERE: IWA WDSO: Stack trace:  
javax.security.auth.login.LoginException: Message stream modified (41)  
at com.sun.security.auth.module.Krb5LoginModule.attemptAuthentication  
  (Krb5LoginModule.java:696)  
at com.sun.security.auth.module.Krb5LoginModule.login(Krb5LoginModule.java:542)  
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)  
at sun.reflect.DelegatingMethodAccessorImpl.invoke  
  (DelegatingMethodAccessorImpl.java:25)  
at java.lang.reflect.Method.invoke(Method.java:597)  
at javax.security.auth.login.LoginContext.invoke(LoginContext.java:769)  
at javax.security.auth.login.LoginContext.access$000(LoginContext.java:186)  
at javax.security.auth.login.LoginContext$4.run(LoginContext.java:683)  
at java.security.AccessController.doPrivileged(Native Method)  
at javax.security.auth.login.LoginContext.invokePriv(LoginContext.java:680)  
at javax.security.auth.login.LoginContext.login(LoginContext.java:579)  
at org.forgerock.jaspi.modules.iwa.wdsso.WDSO.serviceLogin(WDSO.java:601)  
at org.forgerock.jaspi.modules.iwa.wdsso.WDSO.initWindowsDesktopSSOAuth  
  (WDSO.java:560)  
at org.forgerock.jaspi.modules.iwa.wdsso.WDSO.process(WDSO.java:139)  
at org.forgerock.jaspi.modules.iwa.IWAModule.validateRequest(IWAModule.java:107)  
at org.forgerock.openidm.jaspi.modules.IWAModule.validateRequest  
  (IWAModule.java:105)  
at org.forgerock.openidm.jaspi.modules.IWAPassthroughModule.validateRequest  
  (IWAPassthroughModule.java:114)  
at org.forgerock.openidm.jaspi.modules.IDMServerAuthModule.validateRequest  
  (IDMServerAuthModule.java:139)  
at org.forgerock.jaspi.container.ServerAuthContextImpl.validateRequest  
  (ServerAuthContextImpl.java:177)  
at org.forgerock.jaspi.filter.AuthNFilter.doFilter(AuthNFilter.java:162)  
====
```

Missing or incorrectly named keytab file

If the keytab file is absent or does not match the default keytab file name that Identity Connect expects (`identityConnect.HTTP.keytab`), an error similar to the following is output to the `openidm0.log.*` files:

```
====  
org.forgerock.jaspi.modules.iwa.wdsso.WDSO verifyAttributes  
SEVERE: IWA WDSO: Key Tab File does not exist  
====
```

This error should be resolved when you copy the keytab file to the `path/to/salesforceIdConnect/security` directory (or when you rename the keytab file with the correct name). The new keytab file will be picked up automatically - there is no need to restart Identity Connect.

DNS issue with the connection URL

Problems with the DNS record of the Identity Connect host can result in an error similar to the following:

```
SEVERE: IWA WDSSO: Authentication failed with GSSException. Failure unspecified
at GSS-API level (Mechanism level: checksum failed)
```

To check whether there is a DNS issue, inspect the Kerberos tickets on the windows client from which the authentication was initiated. For example:

```
PS C:\Users\Administrator> klist
...
Cached Tickets: (2)

#0> Client: Administrator @ AD.EXAMPLE.COM
Server: krbtgt/AD.EXAMPLE.COM @ AD.EXAMPLE.COM
KerberosTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
Start Time: 9/19/2016 3:10:44 (local)
End Time: 9/19/2016 13:10:44 (local)
Renew Time: 9/26/2016 3:10:44 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
Cache Flags: 0x1 -> PRIMARY
Kdc Called: AD

#1> Client: Administrator @ AD.EXAMPLE.COM
Server: HTTP/myserver001.example.com @ AD.EXAMPLE.COM
KerberosTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 9/19/2016 3:10:44 (local)
End Time: 9/19/2016 13:10:44 (local)
Renew Time: 9/26/2016 3:10:44 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
```

In this sample output, the connection URL for Identity Connect should be `connect.ad.example.com` and not `myserver001.example.com`. Examining the DNS configuration revealed that the DNS record for Identity Connect was an A record (`myserver001.example.com`) and not the required CNAME record (`connect.ad.example.com`). For more information on these record types, see [this article](#).

14.2. Recreating the Identity Connect Repository

Databases can become corrupt for a number of reasons. In the event of a corrupt database, you might need to recreate the Identity Connect repository.

Caution

Be aware that recreating the repository results in the loss of the following information:

- Any existing Salesforce Permission Set to Active Directory Group mappings
- Any existing Salesforce Group to Active Directory Group mappings

- The details of any Active Directory or Salesforce user accounts that were added to the list of Ignored Users
This information must be recreated after the repository has been recreated.

The following procedure explains how to recreate an OrientDB or JDBC repository.

1. Stop Identity Connect, if it is running.

- To stop Identity Connect on UNIX-like systems, run the shutdown script, located in the install directory.

```
$ cd /path/to/salesforceIdConnect
$ ./shutdown.sh
Stopping OpenIDM (91957)
```

- To stop Identity Connect on Windows systems, stop the OpenIDM application in the Windows Task Manager, or type `shutdown` in the Felix console that opened when you started Identity Connect.

2. Delete and recreate the repository schema.

- For an OrientDB repository, delete the `db/openidm` directory and all its subdirectories.
- For an external JDBC repository, drop the `openidm` schema and re-create it, as described in Section 11.1, "Setting Up Identity Connect With MySQL".

3. Restart Identity Connect.

- To restart Identity Connect on UNIX-like systems, run the startup script, located in the install directory.

```
$ cd /path/to/salesforceIdConnect
$ ./startup.sh
```

- To restart Identity Connect on Windows systems, run the `startup.bat` script in the installation directory.

4. Log in to the administrative interface (<https://hostname.domain:8443/admin/>).

5. Select the Salesforce.Org tab.

6. For each configured Salesforce organization, select the Mapping tab, and perform the following tasks.

- a. Select Permission Set to AD Group and recreate any previously defined permission set to group mappings.

Click Update Now to refresh any cached data.

- b. Select SF Group to AD Group and recreate any previously defined group mappings.

Click Update Now to refresh any cached data.

7. Select the Sync tab.

Add any users who had previously been marked as ignored to the list of Ignored Users.

8. Click Sync Now to perform a reconciliation. Depending on the size of the database and system hardware, this process might take a number of minutes.

14.3. General Troubleshooting

- Occasionally, during startup, the following Null Pointer Exception is observed in the logs:

```
SEVERE: Bundle: org.forgerock.openidm.servlet-registrator [80]
[org.forgerock.openidm.servletfilter] The activate method has thrown an exception
java.lang.NullPointerException
```

This error occurs when the servletfilter OSGi bundle fails to activate, due to an NPE within the pax-web-jetty-bundle-3.0.0 OSGi bundle.

If you observe this error, restart Identity Connect.

- A warning similar to the following is observed in the logs:

```
2014-09-15 17:03:44:169 WARN Invalid tag format detected: 9147529c [restlet]
```

This is a harmless message, originating in the bundled Restlet code. The warning is expected, and can be ignored.

Identity Connect Glossary

reconciliation	The process of analyzing data on a target system to determine its consistency with the data on a source system.
synchronization	The process of modifying data on a target system to maintain consistency with the data on a source system.

Index

C

connections, 18

G

Getting started, 4

M

mapping data, 34

R

Repository database

 SQL database, 91

 Table names, 92