



Connector Configuration: AD, LDAP, JDBC

IdentityIQ Version: 5.5

The Direct connector offerings have been expanded in IdentityIQ versions 6.0 and 6.1 while the Governance and Gateway connectors have been replaced or are being phased out, rendering much of this white paper obsolete for those newer product releases. Customers using version 5.5 can still use this document for guidance in configuring the connectors they need, but customers using IdentityIQ versions 6.0+ should consult the Direct Connectors Admin and Configuration Guide for Direct connector installation instructions. That guide is included in the IdentityIQ 6.0+ zip packages.

This white paper will not be updated for future IdentityIQ releases.

This document gives an overview of IdentityIQ's connectors and describes the steps for configuring some of the most frequently used connectors. Specifically, it discusses Active Directory, LDAP, and JDBC connectors. It was written using IdentityIQ Version 5.5, but some of the content may apply to both earlier and later versions of the product.

Table of Contents

IdentityIQ Connectivity Overview	4
Connector Basics	4
How IdentityIQ Selects the Provisioning Mechanism	4
How the Connectors Work	4
Governance Connectors	4
Direct Connectors	5
Gateway Connectors	6
Agent Connectors	8
MCSC Types for Connectors	8
Viewing the Available Connectors	9
Connector Licensing	10
Connector Selection	10
Implementing the Connectors	11
Active Directory	12
AD Governance Connector	13
AD Direct Connector	15
Install IQService	15
Configure the AD Application	16
Active Directory Provisioning Policy	18
AD Gateway Connector	18
Pre-Installation Activities	19
Active Directory Connector Administrator Account	19
Microsoft Visual C++ Runtime Version	19
Connector Manager Installation	19
Connector Installation	20
Connector Gateway Installation	22
Application Definition	24
Troubleshooting	26
Custom Attributes	27
LDAP	28
LDAP Governance Connectors	29

LDAP Direct Connectors	31
Viewing the Connector Configurations	31
LDAP Connector Customizations	33
Available Customization Parameters.....	34
Configuration Steps	37
LDAP Gateway Connectors	39
Pre-Installation Activities.....	40
Connector Manager Installation.....	40
Connector Installation	41
Connector Gateway Installation	43
Application Definition.....	46
JDBC	49
JDBC Direct Connector	49
Connector Rules	53
Build Map Rule	53
MergeMaps Rule	53
Map to ResourceObject Rule (Transformation Rule)	54
JDBC Provision Rule	54
JDBC Governance Connector.....	56

IdentityIQ Connectivity Overview

IdentityIQ's functionality is dependent on its ability to collect data from and, in many cases, send data to other applications. This communication is managed through a combination of application connectors and integration configurations.

Connector Basics

IdentityIQ makes use of several different types of connectors. Connectors are commonly grouped by the ways in which they can communicate with IdentityIQ. There are:

- read-only connectors that can only communicate data *into* IdentityIQ from an external application
- read-write connectors that can read data from external applications *and* write data out to them

The read-write category further subdivides into 3 connector types based on their implementation infrastructure. This chart indicates the types of connectors that are included in each category.

Read Only	Read-Write		
Governance	Gateway	Agent	Direct

How IdentityIQ Selects the Provisioning Mechanism

IdentityIQ determines how to write to an external application (Application A) by examining the following information in this order and executing the first option for which the condition is met:

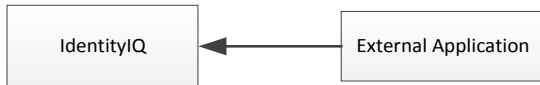
1. Does the FeaturesString on Application A's connector specify "Provisioning"? If so, write through that connector.
2. Does the ManagedResource list on another application's (Application B) connector specify Application A as a managed resource? If so, write through Application B's connector.
3. Does the ManagedResource list on an integration config specify Application A as a managed resource? If so, write through that integration config's executor.
4. Is there an integration config with an attribute: `<entry key='universalManager' value='true'/>`? If so, write through that integration config's executor as the default.

NOTE: Integration Configs are a separate construct used for writing to provisioning integration systems and to help desk systems. These are outside the scope of this document but are mentioned here to provide a complete picture of IdentityIQ's communications with other applications.

How the Connectors Work

Governance Connectors

Governance connectors are very simple in design; they make a direct read-only connection to the external application through the connection parameters specified on the Application Definition.



The currently available Governance Connectors are listed here.

Governance Connectors
Active Directory (for IdentityIQ versions 5.5p1 and earlier)
IBM Lotus Domino
LDIF
Microsoft SharePoint
Microsoft SQL Server
Oracle
PeopleSoft
SAP
SAP HR/HCM
SAP Portal – UMWebService
Unix
VMS
Mainframe
RACF Full
TopSecret
Delimited File
Logical
RuleBasedFileParser
RuleBasedLogical

Direct Connectors

Direct connectors are read-write connectors that allow IdentityIQ and the external application to send data directly between them in both directions. When read and write capabilities are needed for applications that have these connectors available, they are the most efficient and best choice to implement.



The current set of direct connectors is listed below. Over time, more of these will be made available.

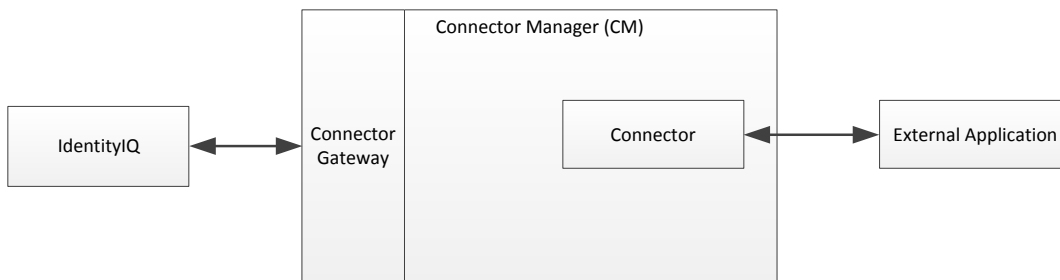
Direct Connectors
ADAM – Direct
JDBC (replaces Governance connector in 5.2p1)
Novell Edirectory – Direct
OID – Direct
OpenLDAP – Direct
SunOne – Direct
Tivoli – Direct

Google Apps
Webex
Salesforce
Active Directory (replaces Governance connector in 5.5p2+)

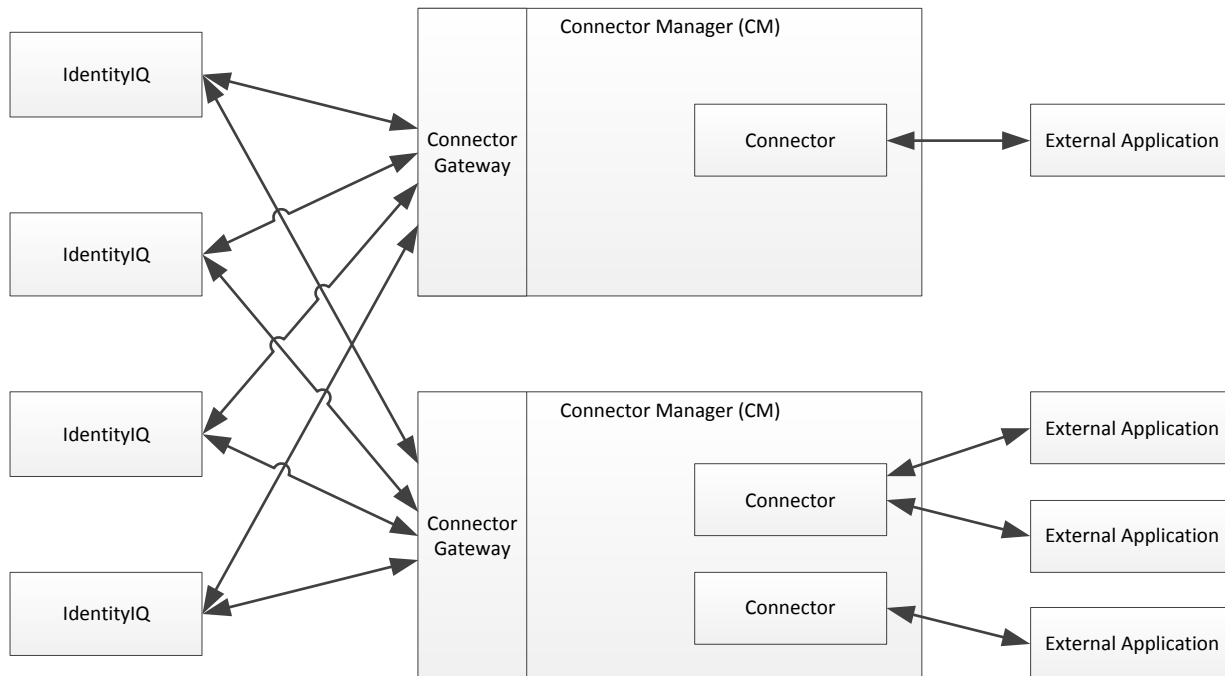
Gateway Connectors

Gateway connectors implement a multi-component infrastructure for managing communications with external systems. With the Gateway connectors, the connection to the external application is managed by a Connector Manager and a connector. Each Connector Manager can only accept only one incoming connection, so the Connector Gateway acts as an intermediary, accepting concurrent requests from one or more IdentityIQ instances and forwarding those requests to the Connector Manager. Prior to IdentityIQ release 5.5p3, the Connector Gateway was installed as a separate application from the Connector Manager, but it has now be incorporated into the Connector Manager so the Gateway does not have to be installed separately. The Connector Manager hosts the connector, which communicates directly with the external application (managed system). Each connector can communicate with only one managed system type, though it can connect to multiple instances of that system type through separately defined managed system configurations. In most cases, a single CM instance can address multiple connectors.

The basic configuration for Gateway connectors looks like this:



When more than one IdentityIQ application server is in place and more than one application requires a read-write connection, the configuration might look something like this:

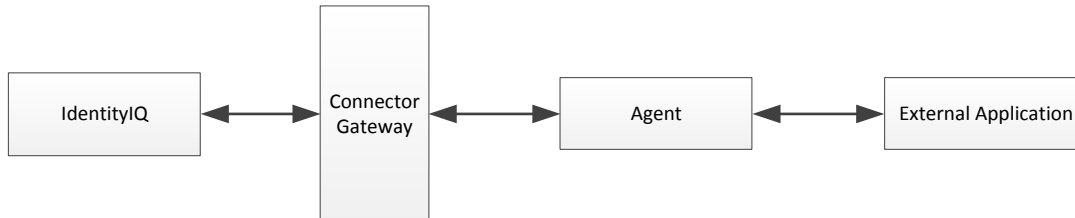


These are the currently available Gateway Connectors.

Gateway Connectors
Ace Server
Active Directory Full
AIX
ADAM - Gateway
ITSM
Linux
Lotus Notes
Microsoft SQL Server Full
Oracle Applications
Oracle Full
PeopleSoft PM
Remedy
SAP EP
SAP HR
SAPR3
Solaris
SunOne - Gateway
Sybase
Tivoli - Gateway
Windows Local Full

Agent Connectors

The targeted systems for Agent connectors are centralized mainframe security systems; Agents are the simplest and most secure way to connect to those systems. Like the Gateway connectors, Agents communicate with IdentityIQ through the Connector Gateway. In Agent connectors, the functionality of the Connector Manager is contained within the Agent, so the Connector Manager is not required.



These are the Agent Connectors for IdentityIQ.

Agent Connectors
ACF2
AS400
RACF Full
TopSecret Full

MCSC Types for Connectors

The IdentityIQ Application Configuration for some connector types requires the specification of an **MCSC Type** parameter (MCSC = Managed System Configuration Set). This table indicates the MCSC type appropriate to each connector for which it is a required field:

Managed System	MCSC Type	Name of the Connector
Active Directory	Win2000	Connector for Microsoft Active Directory
LDAP ADAM	LDAPADAM	Connector for LDAP Directories
LDAP SunOne	LDAPIP	Connector for LDAP Directories
LDAP Tivoli	LDAPT16.0	Connector for LDAP Directories
Lotus Notes	Notes	Connector for Lotus Notes
Windows	WinLocal	Connector for Microsoft Windows Local Security
Novell	NetWare	Connector for Novell NetWare
Microsoft SQL Server	MSSQL	Connector for Microsoft SQL Server
Oracle	Oracle	Connector for Oracle Server
Sybase ASE	Sybase	Connector for Sybase Adaptive Server Enterprise
IBM DB2	DB2	Connector for DB2 Universal Database
SAP Solution	SAPR3	Connector for SAP Solutions
SAP EP	SAPEP	Connector for SAP Enterprise Portal
Oracle Applications	OraAppl	Connector for Oracle Applications

PeopleSoft	PSoft	Connector for PeopleSoft
HR Applications	SAPHR	Connector for HR Applications
Linux	Linux	Connector for Linux
Solaris	Solaris26	Connector for Solaris
AIX	AIX42	Connector for AIX
HP-UX	HP-UX10	Connector for HP-UX
AS-400	AS400	Connector for AS/400
NIS	NIS	Connector for NIS
RACF	RACF	Connector for RACF
CA-ACF2	ACF2	Connector for CA-ACF2
CA-TopSecret	TSS	Connector for CA-Top Secret
RSA AM	AceServer	Connector for RSA Authentication Manager
ITSM	ITSM	Connector for BMC Remedy IT Service Management Suite
Remedy	Remedy	Connector for Remedy AR System

Viewing the Available Connectors

The list of available connectors can be retrieved from the Connector Registry within the IdentityIQ Debug Pages. Select **Configuration** in the Objects list and click **List**.

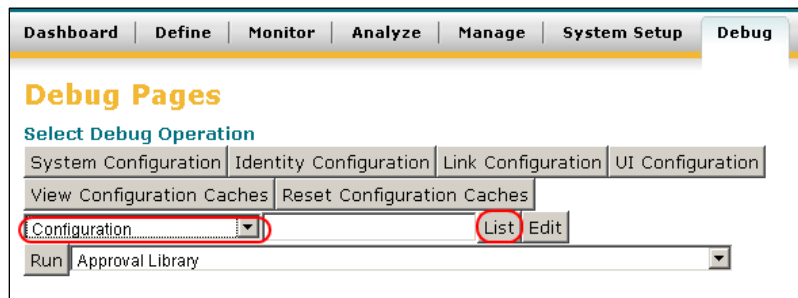


Figure 1: Debug Pages

Select **ConnectorRegistry** to view the XML for all the connectors.



Figure 2: Configuration Objects List

The **featuresString** value on each connector indicates the functionality that connector is capable of providing; when **PROVISIONING** is specified in the **featuresString**, the connector is a write-capable connector. The attribute “<entry key=“MscsType” value=“[MSCS-Type-Name]”/>” tells the name to specify for that connector’s MSCS Type value (also listed in the previous section here).

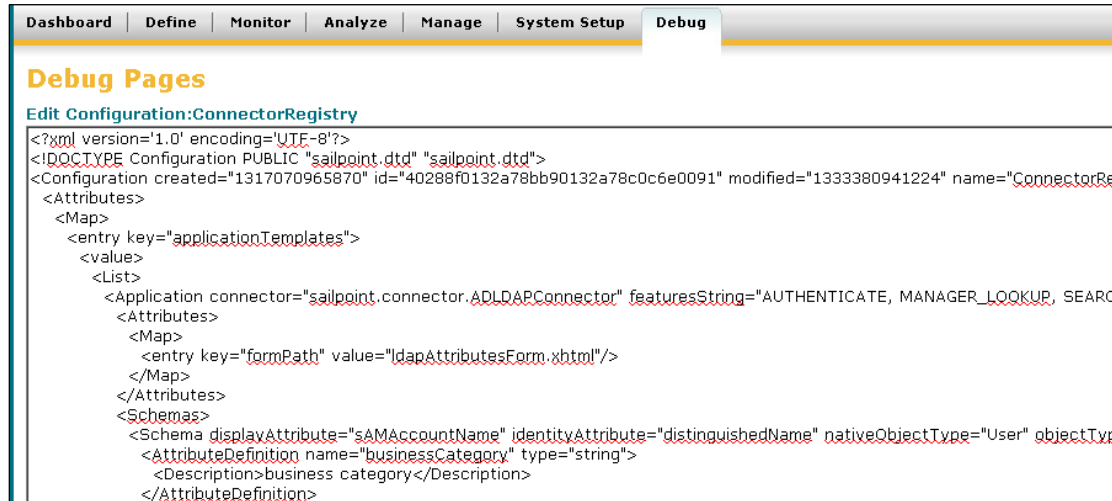


Figure 3: Connector Registry XML

The out-of-the-box connector specifications can also be found in the ConnectorRegistry.xml file in the [IdentityIQ Installation Directory]/WEB-INF/bin directory.

Connector Licensing

All IdentityIQ customers are automatically licensed to use any of the Governance connectors. However, the Direct, Gateway, and Agent connectors are separately licensed with the product’s Provisioning Engine. Many of the Direct connectors began as Governance connectors and were modified to add the Provisioning capabilities. Customers who have not licensed the Provisioning Engine may use these connectors for *reading* data (aggregation) but are not permitted to implement the provisioning features of these connectors without purchasing a Provisioning Engine license.

Connector Selection

Often there is more than one connector that can communicate with a single external application, which may raise questions as to which one is the best choice. When multiple connectors exist for a single application, they are always of different types. The “best” choice is dictated by the needs (and license limitations) of the organization. More information on connector selection is provided in the introductory section for each application within this document.

Implementing the Connectors

The remaining chapters in this document discuss the steps required to implement the various connectors. The document is divided by application, with each application's section describing implementation steps for all connectors available for it.

NOTE: The scope of this document is limited to getting the connectors installed and configured for successful communication between the external application and IdentityIQ. Specific usage needs, such as account group tracking, password maintenance, etc. may require further configurations that may not be covered in this document.

Active Directory

IdentityIQ offers multiple connector options for Active Directory; the set of available connectors depends on the installed version of IdentityIQ.

Connector Name	Type	IdentityIQ Versions
Active Directory	Governance	5.5p1 and earlier
Active Directory Full	Gateway	5.2, 5.5 all patches
Active Directory	Direct	5.5p2 and later

The recommendation on which connector to use depends on the IdentityIQ release installed and whether automated provisioning will be done to AD.

IdentityIQ Release	Automated Provisioning?	Recommendation
5.2	No	Governance (read-only) AD connector
	Yes	Gateway (read/write) AD connector; better performance will be obtained by upgrading to Release 5.5 and using delta aggregations
5.5	No	Governance AD connector
	Yes	Gateway AD connector and delta aggregations
5.5p2+	Yes/No	Direct AD Connector (unless require delta aggregations or after scripts)
	Yes	Gateway AD Connector (if require delta aggregations or after scripts)

The Gateway connector is designed to run a full initial aggregation and then rely on the connectors' interceptors to keep data in sync between IdentityIQ and the managed system through delta aggregation. In this model, there should never be a need to repeat a full aggregation with this connector. The delta aggregation feature, however, is only available for IdentityIQ Release 5.5 and later.

The Gateway connector incurs a significant performance penalty over the Governance connector on aggregations, so installations running on Release 5.2 and not requiring the Gateway connector's write capabilities will experience better aggregation performance by implementing the read-only Governance connector. Because of the improved performance offered by delta aggregation, installations wanting to make use of the provisioning capabilities of the Gateway connector are strongly encouraged to migrate to Release 5.5.

Beginning with release 5.5p2, the Governance connector was converted to a Direct (read-write) connector. Use of this connector for writing still requires the installation to be licensed for provisioning, but this addition of write capabilities introduces a second provisioning connector option. The Direct connector does not support delta aggregations or after scripts at this time (coming in version 6.0) but its full aggregations are typically faster than the Gateway connector's full aggregations. Selection between these two connectors for 5.5p2+ installations depends on whether provisioning is being implemented and on whether delta aggregations or after scripts are required.

AD Governance Connector

The Governance Connector for AD is available in IdentityIQ versions 5.5p1 and earlier. Configure the Active Directory application to use a Governance Connector following these steps.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.

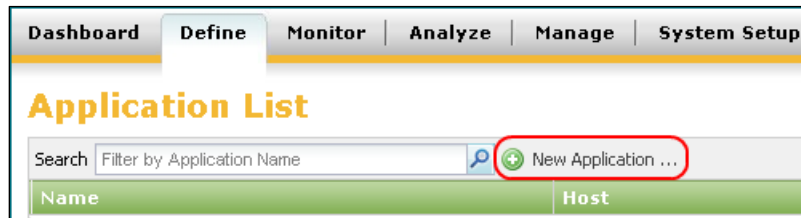


Figure 4: Create New Application

3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select **Active Directory** as the **Application Type**. This tells IdentityIQ to use the AD Governance Connector.

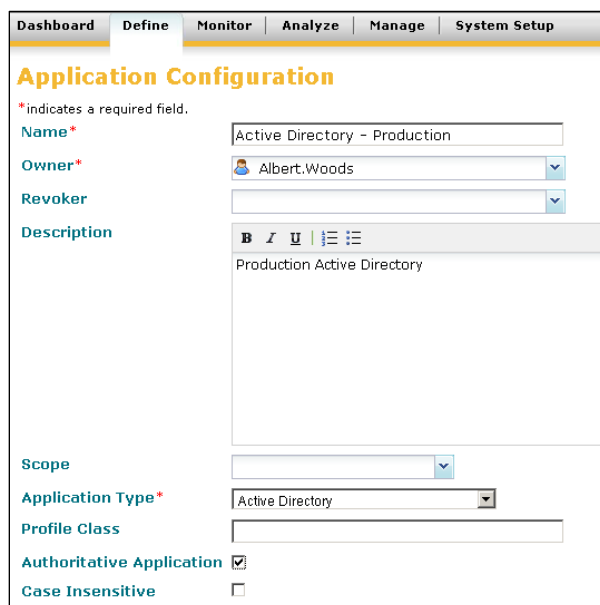
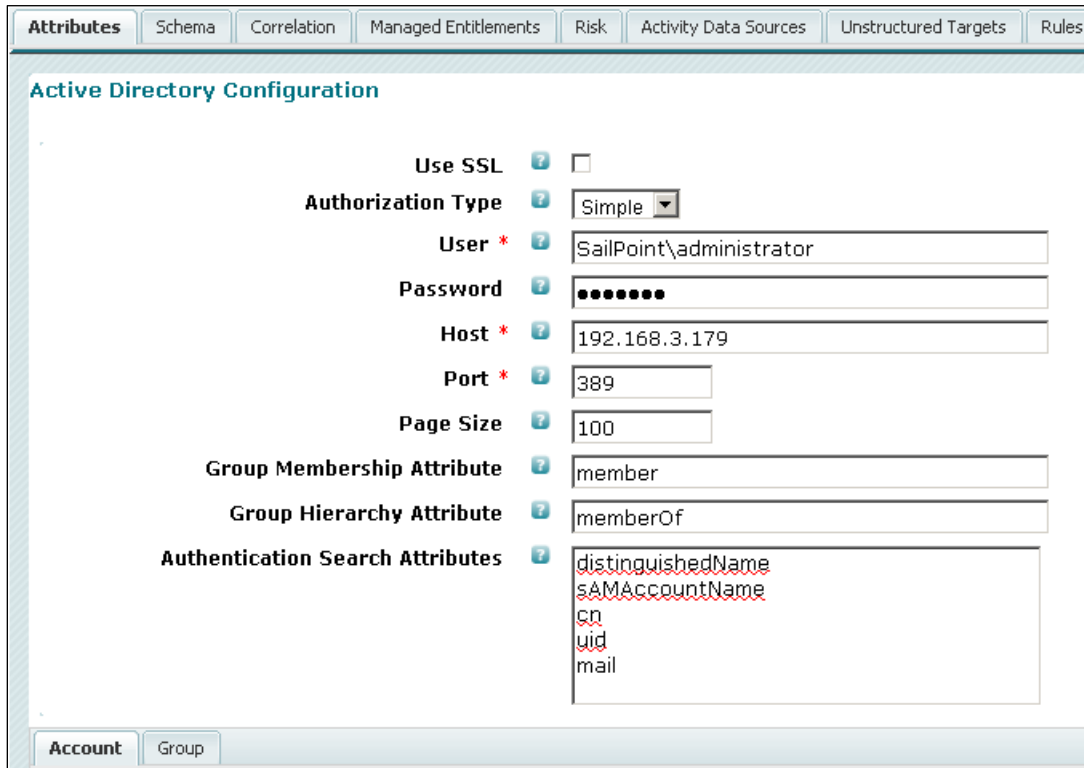


Figure 5: Active Directory connector specification

5. On the **Attributes** tab, specify the connection configuration attributes:
 - **Use SSL:** Select if the connection between the IdentityIQ host and the AD host uses SSL (appropriate client and server certificates will be required)
 - **Authorization Type:** Simple or None; choosing None bypasses authorization and signs on as an anonymous user (which may not be permitted by your AD setup); choosing Simple authenticates with the User and Password provided in the next parameters
 - **User:** account used in making the connection to AD in DomainName\UserName format; required
 - **Password:** the password for the connection user account

- **Host:** the hostname or IP Address of the AD instance's host machine; required
- **Port:** the TCP/IP Port on which the AD instance is listening; required (default is 389 for non-SSL or 636 for SSL)
- **Group Membership Attribute:** member



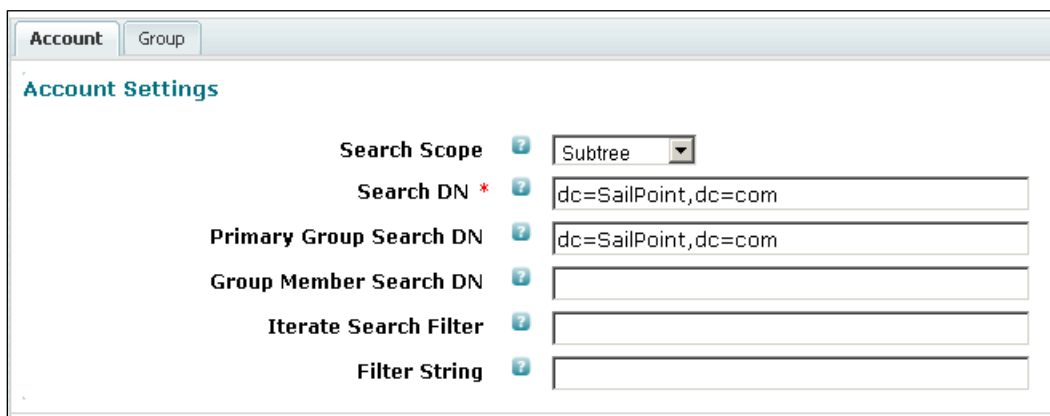
The screenshot shows the 'Active Directory Configuration' form within a software interface. The form has tabs at the top: 'Attributes', 'Schema', 'Correlation', 'Managed Entitlements', 'Risk', 'Activity Data Sources', 'Unstructured Targets', and 'Rules'. The 'Attributes' tab is selected. The form contains the following fields and values:

- Use SSL:** ☐
- Authorization Type:** Simple (dropdown menu)
- User:** SailPoint\administrator
- Password:** [Redacted]
- Host:** 192.168.3.179
- Port:** 389
- Page Size:** 100
- Group Membership Attribute:** member
- Group Hierarchy Attribute:** memberOf
- Authentication Search Attributes:** distinguishedName, sAMAccountName, cn, uid, mail

At the bottom of the form, there are two tabs: 'Account' and 'Group'.

Figure 6: Active Directory Configuration

- Specify the **Search DN** (the DN from which account searches should begin) as well as any of the other optional search parameters. In AD, a user's primary group is not listed in the list of groups in the Member attribute like other groups are and is only found through a follow-up query using the **Primary Group Search DN** as its starting point.

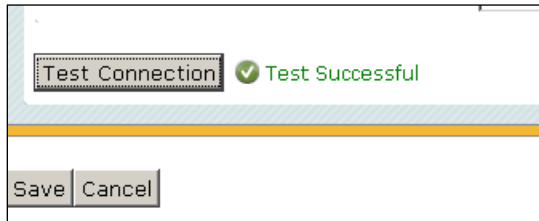


The screenshot shows the 'Account Search Settings' form within a software interface. The form has two tabs: 'Account' and 'Group'. The 'Account' tab is selected. The form contains the following fields and values:

- Search Scope:** Subtree (dropdown menu)
- Search DN:** dc=SailPoint,dc=com
- Primary Group Search DN:** dc=SailPoint,dc=com
- Group Member Search DN:** [Empty]
- Iterate Search Filter:** [Empty]
- Filter String:** [Empty]

Figure 7: Account Search Settings

7. Click **Test Connection** (at the bottom of the page) to verify whether IdentityIQ can connect to the Active Directory instance with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successful or not. Non-success messages contain error information to help diagnose the connection problem.



8. Click **Save** to save the application definition.

AD Direct Connector

The AD Direct Connector was released with the IdentityIQ 5.5p2 patch release. It is a modification to the AD Governance connector that was available in releases prior to 5.5p2 and supports these functions in addition to the read functionality of the Governance connector:

- Create/Delete Users
- Manage Terminal Services, Dial-in Attributes
- Add custom attributes to provisioning policy to set the extended attributes while creating users
- Manage Exchange 2007, Exchange 2010
- Enable, Disable, Unlock, Reset Password for Users
- Add/Remove entitlements

The AD Direct connector relies on IQService for its provisioning functionality. IQService is a native Windows service that enables IdentityIQ to participate in a Windows environment and access information only available through Win32 APIs.

Install IQService

IQService must be installed and registered to allow the connector to provision to Active Directory, aggregate Terminal services attributes, collect information from the Windows Event Logs, or load local Windows users or groups.

Some customers may have previously installed and registered this service to support file permission management. If it is already installed, it must be upgraded to the newest version. First stop and remove the service and then follow the instructions for installing the new version.

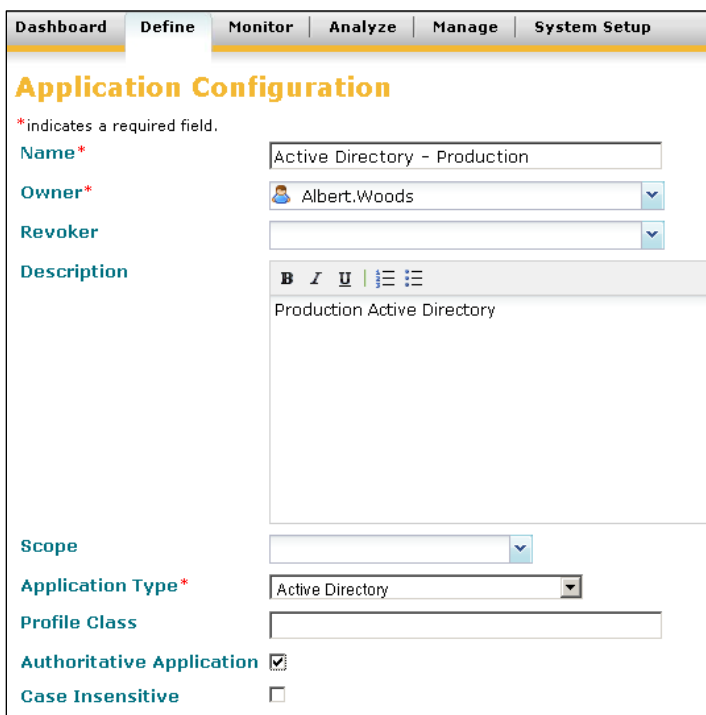
The steps to install IQService are found in the file 5_5_Active_Directory_Technical_Bulletin.pdf in the section titled **Install and Register the IQService for Windows**. This file is found in the [IdentityIQ installation directory]/doc/pdf directory once the 5.5p2 patch is unjarred. This bulletin also includes system requirements for the host machine where the IQService will be installed, as well as commands for stopping and removing the service.

NOTE: The zip file that contains the IQService executable and required DLLs can be found in the [IdentityIQ installation directory]/WEB-INF/bin/win directory.

Configure the AD Application

Configuring the Active Directory application to use the Direct Connector parallels the process for configuring the Governance connector but requires some additional parameters that did not apply to the Governance connector.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.
3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select **Active Directory** as the **Application Type**. This tells IdentityIQ to use the AD Direct Connector.



The screenshot shows the 'Application Configuration' window with the 'Define' tab selected. The window has a header with tabs: Dashboard, Define, Monitor, Analyze, Manage, and System Setup. The main content area is titled 'Application Configuration' and includes a note: '*indicates a required field.' The form fields are as follows:

- Name***: Text input field containing 'Active Directory - Production'.
- Owner***: Dropdown menu showing 'Albert.Woods'.
- Revoker**: Dropdown menu (empty).
- Description**: Rich text editor containing 'Production Active Directory'.
- Scope**: Dropdown menu (empty).
- Application Type***: Dropdown menu showing 'Active Directory'.
- Profile Class**: Text input field (empty).
- Authoritative Application**: Checkmark box (checked).
- Case Insensitive**: Checkmark box (unchecked).

Figure 8: Active Directory connector specification

5. On the **Attributes** tab, specify the connection configuration attributes. These differ slightly from the values required for the Active Directory Governance connector.
 - **IQService Host**: IP or hostname of the host where IQService is installed (Domain Controller must be accessible from this host computer)
 - **IQService Port**: port on which IQService is listening (default is 5050; examine the Windows Registry entry to verify the port number)
 - **Use SSL**: Select if the connection between the IdentityIQ host and the AD host uses SSL (appropriate client and server certificates will be required)
 - **Authorization Type**: Simple or None; choosing None bypasses authorization and signs on as an anonymous user (which may not be permitted by your AD setup); choosing Simple authenticates with the User and Password provided in the next parameters

- **User:** The user (administrator) name used to make the connection, written in DomainName\UserName format; required
- **Password:** the password for the connection user account
- **Host:** the hostname or IP Address of the AD instance's host machine; required
- **Port:** the TCP/IP Port on which the AD instance is listening; required (default is 389 for non-SSL or 636 for SSL)
- **Group Membership Attribute:** member
- **Exchange Version:** version of Microsoft Exchange, if it is to be managed by the application

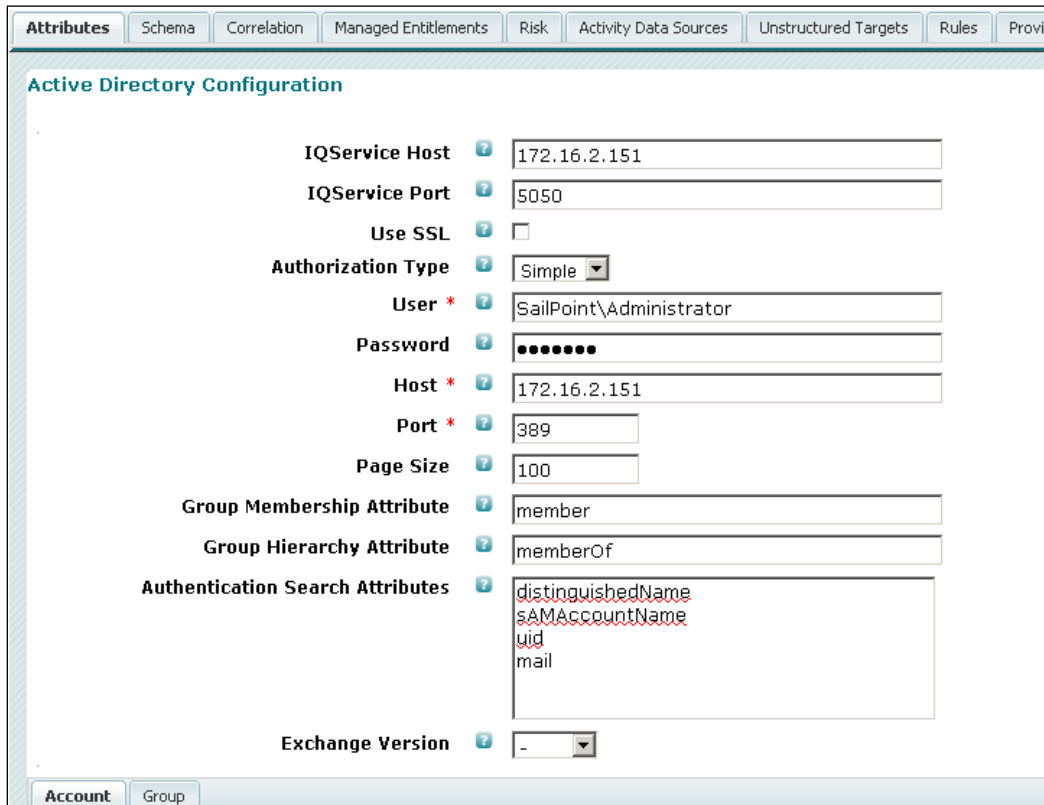


Figure 9: Active Directory Direct Configuration

- Specify the **Account Settings** search information:
 - **Search Scope:** Scope of search within the directory
 - **Search DN:** DN from which to start a directory search for accounts
 - **Primary Group Search DN:** DN from which to start a search for a user's primary group . In AD, a user's primary group is not listed in the list of groups in the Member attribute like other groups are and is only found through a follow-up query using the **Primary Group Search DN** as its starting point.
 - **Group Member Search DN:** DN from which to start a search when resolving a user's group memberships

If applicable, specify the Group Settings. If groups are not being managed by the application, this is not necessary. If they are managed by the application and no group settings are specified, the Account Settings values will be used.

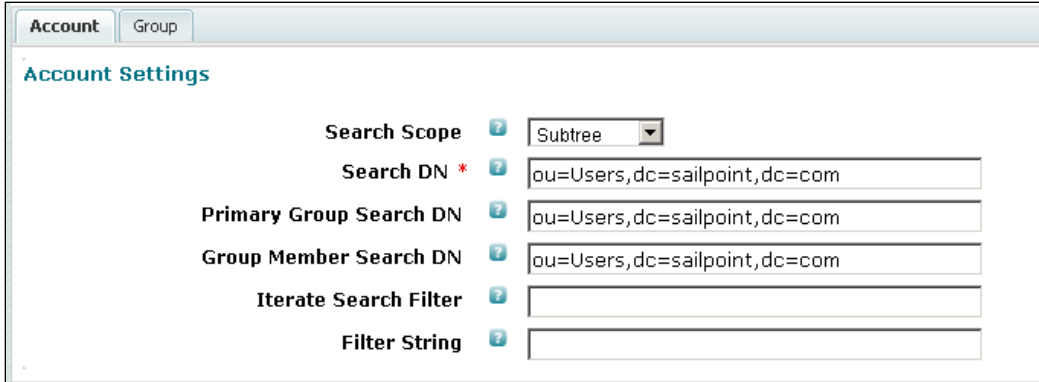


Figure 10: Account Search Settings

7. Click **Test Connection** (at the bottom of the page) to verify whether IdentityIQ can connect to the Active Directory instance with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successful or not. Failure messages contain error information to help diagnose the connection problem.

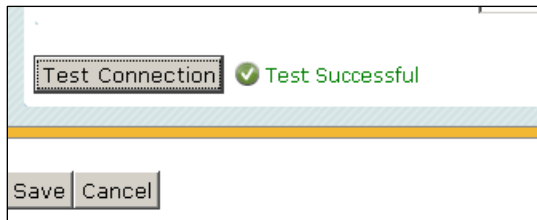


Figure 11: Successful Test Connection

8. Click **Save** to save the application definition.

Active Directory Provisioning Policy

The Active Directory Direct connector is pre-configured with an account creation provisioning policy that includes the commonly-used attributes that need to be set when creating an account. This field list can be modified as needed.

If Terminal Services are being managed by this application, there are a few more attributes that must be added to the schema and to the provisioning policy. These are indicated and described in the **Active Directory Connector – Provisioning Policy** section of the 5_5_Active_Directory_Technical_Bulletin.pdf.

AD Gateway Connector

Setting up Active Directory with a read-write Gateway Connector requires some additional configuration and application-installation steps. These steps should be done in the order specified here.

1. Complete Pre-installation Activities
2. Install the Connector Manager
3. Install the Connector
4. Install the Connector Gateway (only for IdentityIQ 5.5p2 and earlier)

5. Specify the Active Directory Application Configuration in IdentityIQ

Pre-Installation Activities

Additional information about these required pre-installation activities can be found in the Connector Manager installation guide. Installations running IdentityIQ 5.5p3 or later should refer to the 6_0_ConnectorManager_Windows_Install_Guide.pdf file, while those running IdentityIQ versions prior to 5.5p3 should see 5_2_ConnectorManager_Windows_IG.pdf, provided with the connector manager installation files.

Active Directory Connector Administrator Account

Create (or identify) an Active Directory account for the Connector Administrator that will be used for all transactions done through the Connector. This account requires these permissions:

- Member of Domain Admin group (or at least sufficient rights to read and write the different attributes)
- If this connector will manage Microsoft Exchange, the administrator must be part of Organization Management.
- Local administrator on the local system
- Ability to log on as a service on the local system

Microsoft Visual C++ Runtime Version

Connector Manager requires Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package MFC Security Update Version- 8.0.61001, 64 bit- 8.0.61000. Check **Control Panel > Installed Programs** to determine whether the installed version is correct or needs to be upgraded. (**NOTE:** This is a different version than was required by the BMC Services Manager application, so sites upgrading from SM to CM need to ensure that the newer runtime is installed.)

Attempting to run any Connector Manager executable with the incorrect runtime will result in the error message "The application has failed to start because its side-by-side configuration is incorrect." Attempting to open the CMAdmin console with the wrong runtime would cause it to hang at the "Initializing" screen.

If an incorrect Visual C++ runtime is discovered after CM is installed, it is not necessary to reinstall Connector Manager. The two CM utility files ctsprc.exe and ctsqcr.exe can be run manually to create the QUEUE files and MSOFLI file, as documented in the CM installation guide.

Connector Manager Installation

Each Active Directory Connector requires its own dedicated Connector Manager; it cannot share a CM with another AD Connector or with a connector of a different type. This is due to some common parameters required by CM and the AD Connector that will impact other connectors' functionality if deployed on the same CM. Connector Manager may be installed on any Active Directory platform (server or workstation) that is a part of any domain in the Active Directory Domain Tree to be managed.

Install Connector Manager for Windows and any patches that apply to the release. Obtain the installation files from your SailPoint representative. Select the most recent version appropriate to your operating system (32-bit or 64-bit). The installation process includes all the configuration options required to prepare Connector

Manager for use. Encryption can be configured during installation or at a later time, as described in the installation guides.

The table below shows the installation files and patch files required for each of the most recent CM versions.

IdentityIQ Versions	CM Version	Installation File	Patch File
5.2	5.2	ConnectorManager-Windows-xxbit-5.2p1.zip	none
5.5GA – 5.5p2	5.5	ConnectorManager-Windows-xxbit-5.2p1.zip	ConnectorManager-Windows-xxbit-5.5.00-patch.zip
5.5p3+	6.0	ConnectorManager-Windows-xxbit-6.0.zip	none

Installation instructions are found in these documents:

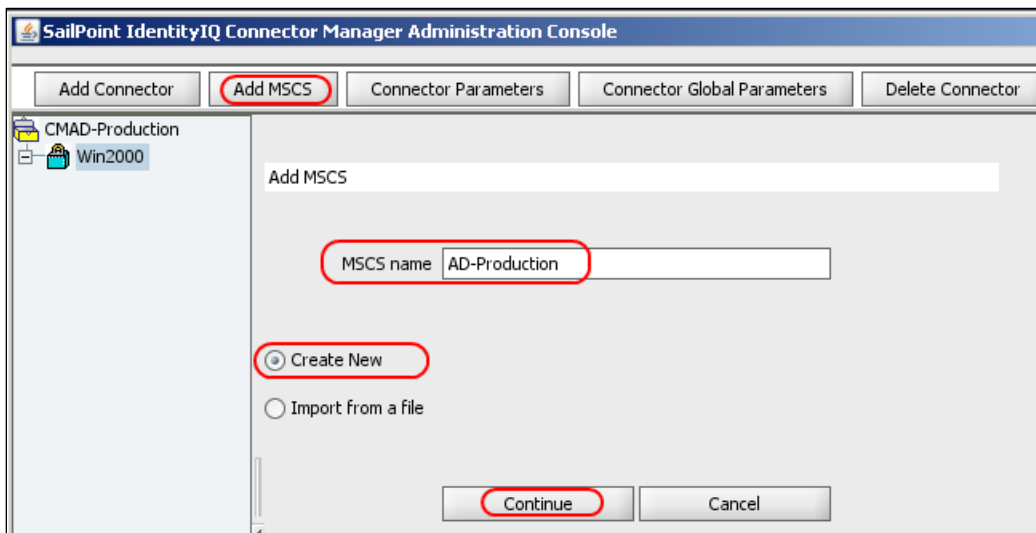
Installation/Patch File	Instruction Document
ConnectorManager-Windows-xxbit-5.2p1.zip	5_2_ConnectorManager_Windows_IG.pdf
ConnectorManager-Windows-xxbit-5.5.00-patch.zip	5_5_ConnectorManager_Install_Guide.pdf
ConnectorManager-Windows-xxbit-6.0.zip	6_0_ConnectorManager_Windows_Install_Guide.pdf

Connector Installation

The Connector for Microsoft Active Directory can manage an entire Active Directory Domain Tree. Complete these steps to install and configure the Connector.

1. Install the Connector for Microsoft Active Directory.
 - Obtain the installation files from your SailPoint representative. Select the most recent version. At the time of this writing, the most recent version is in the file Connector-ActiveDirectory-5.2.zip.
 - The installation guide for installing the connector (and its earliest patches) found in the file: 5_5_Connectors_Install_Guide.pdf.
2. Install any patches that apply to the release.
 - Several patches have been created for this version, most of which are included in the 5.2.zip file. The first patch (5.1.00.200) must be applied using the Connector Manager's addSpConnector.bat script file. Details on this are included in the Connector's installation guide. The other patches must be applied by copying files into the appropriate directories, as noted in the installation guide. When patches overwrite files previously updated by previous patches, the interim patches may be skipped if desired.
 - The Connector-ActiveDirectory-5.2p1.zip file is a separate patch release that brings the connector to version 5.1.00.207, the latest version. This file must also be downloaded and installed following its supplemental directions, which are found in the Doc directory within the zip file.
3. Configure the Connector's MSCS parameters through the Connector Manager Configuration Console.
 - Launch the Connector Manager Configuration Console: **Start -> SailPoint -> Connector Manager -> [CM instance name] > CM Administration Console.**
 - Select the Connector (**Win2000**) in the tree in the left pane and click **Add MSCS.**

- Specify an **MSCS Name** to use for the managed system. This name is assigned here and entered in IdentityIQ to indicate the managed system to which the connection applies; this name is case-sensitive. Select **Create New** and click **Continue**.



SailPoint IdentityIQ Connector Manager Administration Console

Buttons: Add Connector, **Add MSCS**, Connector Parameters, Connector Global Parameters, Delete Connector

Left pane: CMAD-Production, Win2000

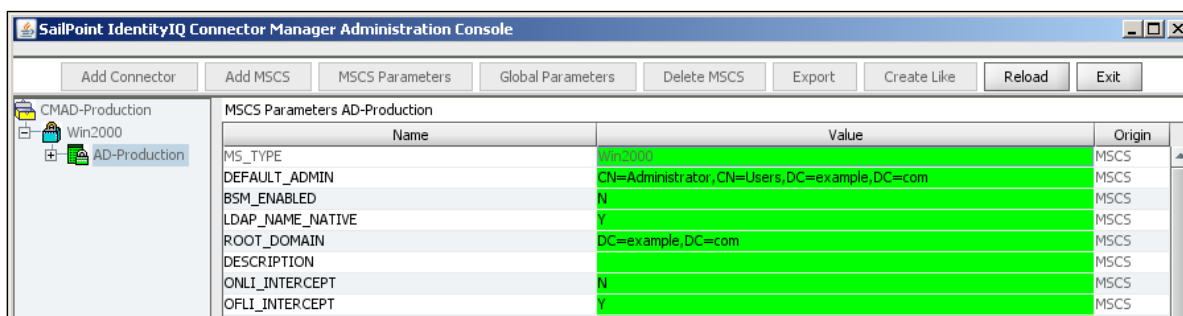
Right pane: Add MSCS

MSCS name: **AD-Production**

Radio buttons: **Create New**, Import from a file

Buttons: **Continue**, Cancel

- Enter the values required for connecting to the Active Directory instance.
 - DEFAULT_ADMIN**: the full Distinguished Name (DN) of the AD Connector Administrator defined in the pre-installation activities.
 - DEFAULT_ADMIN_PASSWORD**: Password of the connector administrator. **NOTE**: After these MSCS parameters are initially saved, the password field is not shown in the parameter display so the password cannot be changed here. If this password is changed in future, ctsadm.exe (found in *cmHome/bin*) must be used to update the password in the CM records. Ctsadm is run from the command line and provides a menu of options when started.
 - ROOT_DOMAIN**: Domain name which needs to be managed, in Distinguished Name format. For remote management, the Connector expects an Active Directory domain to be present in the environment. The connector can remotely manage all the computers from this domain. In case of managing native computer it should be blank.



SailPoint IdentityIQ Connector Manager Administration Console

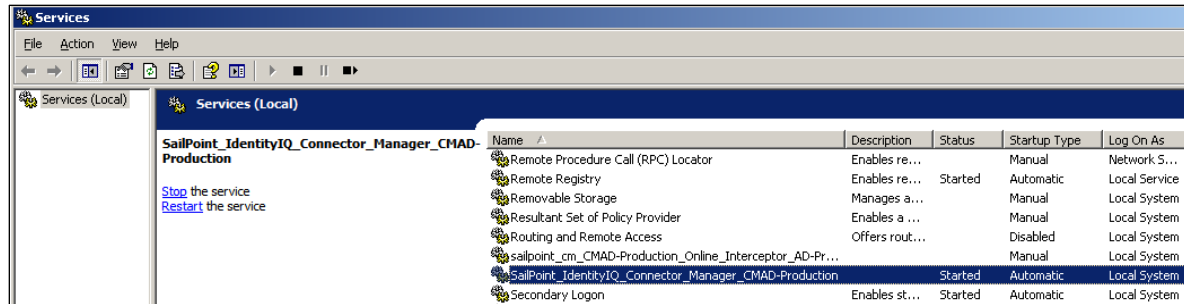
Buttons: Add Connector, Add MSCS, **MSCS Parameters**, Global Parameters, Delete MSCS, Export, Create Like, Reload, Exit

Left pane: CMAD-Production, Win2000, **AD-Production**

Right pane: MSCS Parameters AD-Production

Name	Value	Origin
MS_TYPE		MSCS
DEFAULT_ADMIN	CN=Administrator,CN=Users,DC=example,DC=com	MSCS
BSM_ENABLED	N	MSCS
LDAP_NAME_NATIVE	Y	MSCS
ROOT_DOMAIN	DC=example,DC=com	MSCS
DESCRIPTION		MSCS
ONLI_INTERCEPT	N	MSCS
OFLI_INTERCEPT	Y	MSCS

- Change the **Log on as** account for the CM service (**SailPoint_IdentityIQ_Connector_Manager_[CM instanceName]**) to the connector administrator account.
- Start the CM Service.



Connector Gateway Installation

The Connector Gateway is the intermediary between IdentityIQ and the ConnectorManager. Beginning with CM version 6.0, which pairs with IdentityIQ versions 5.5p3 and later, the Connector Gateway is included within CM and is no longer installed as a separate component. Installations upgrading from CM 5.2 or 5.5 should follow the instructions in the 6.0 Connector Manager Installation guide to remove the external Connector Gateway from the infrastructure:

- Stop the Connector Gateway service.
- Change the IdentityIQ application properties to use the Host and Port details for Connector Manager (instead of the Connector Gateway Host and Port).

The rest of the Connector Gateway installation instructions included here apply only to installations running CM 5.5 or 5.2, which must install the separate Connector Gateway. It can be installed on any machine with network connectivity to both IdentityIQ and the ConnectorManager, but it is strongly recommended that it be installed on the same machine as Connector Manager where possible. One Connector Gateway is required per ConnectorManager.

The Host name (or IP Address) and Port for the Connector Manager must be known to complete the Connector Gateway setup.

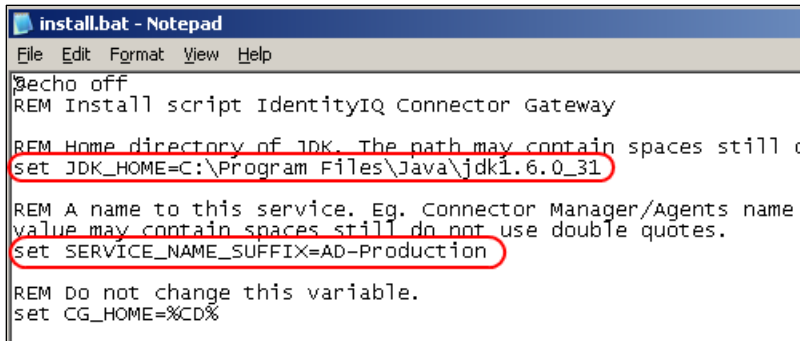
NOTE: These instructions are for installation on a Windows machine, based on the assumption that the Gateway will be co-located with the Connector Manager as recommended. Instructions for installing and configuring on a UNIX machine can be found in the README_UNIX.TXT file within the Connector Gateway's zip file. Other general notes, as well as uninstallation instructions, for CG - Windows can be found in the README_WIN.TXT file inside the Connector Gateway zip file.

NOTE: When installing on Windows, the Connector Gateway only runs on a 32-bit JVM. It will not run as a service on a 64-bit JVM. The 32-bit JDK can be installed on a 64-bit OS.

Complete these steps to install and configure the Connector Gateway:

1. Unzip the Connector Gateway zip file into the directory location where the Connector Gateway will reside. Connector Gateway is part of the IdentityIQ package; install the Connector Gateway version that matches the IdentityIQ version in use. It can be installed in any directory location.
2. Navigate to the Connector Gateway home directory and edit the **install.bat** file to set the JDK directory and the service name:

- **set JDK_HOME=[JDK directory path]** (JDK version 1.5 or 1.6 is required. Download and install this if no JDK is present on the machine.)
- **set SERVICE_NAME_SUFFIX=[service name]** (This assigns the specified name to this Connector Gateway service instance; different Connector Gateway services running on the same machine can be assigned different names.)



```

install.bat - Notepad
File Edit Format View Help
@echo off
REM Install script IdentityIQ Connector Gateway
REM Home directory of JDK. The path may contain spaces still d
Set JDK_HOME=C:\Program Files\Java\jdk1.6.0_31
REM A name to this service. Eg. Connector Manager/Agents name
value may contain spaces still do not use double quotes.
Set SERVICE_NAME_SUFFIX=AD-Production
REM Do not change this variable.
set CG_HOME=%CD%
  
```

Figure 12: Connector Gateway Service Settings

3. Run **install.bat** to create the service.
4. Edit the **init.xml** file in the Connector Gateway's home directory to set the hostname and port of the Connector Manager and the Port for the Connector Gateway. The Connector Gateway's port can be any available port number you choose. This number will be referenced in the IdentityIQ Application Configuration.

```

<ConnectorGateway>

  <SM>
    <!-- Connector Manager/Agents Hostname or IP Address-->
    <hostname>172.16.2.129</hostname>

    <!--Connector Manager/Agents port number-->
    <port>2470</port>

    <!-- Use "AS400" for AS400 system and "MAINFRAME" for Mainframes and
    leave empty in all other cases.-->
    <platform></platform>
  </SM>

  <Server>
    <!-- Connector Gateway port number-->
    <port>5700</port>

    <!-- Delay(in seconds) between two retry attempts while connecting to
    Connector Manager/Agents-->
    <sm_connect_retry>3</sm_connect_retry>

  </Server>
</ConnectorGateway>
  
```

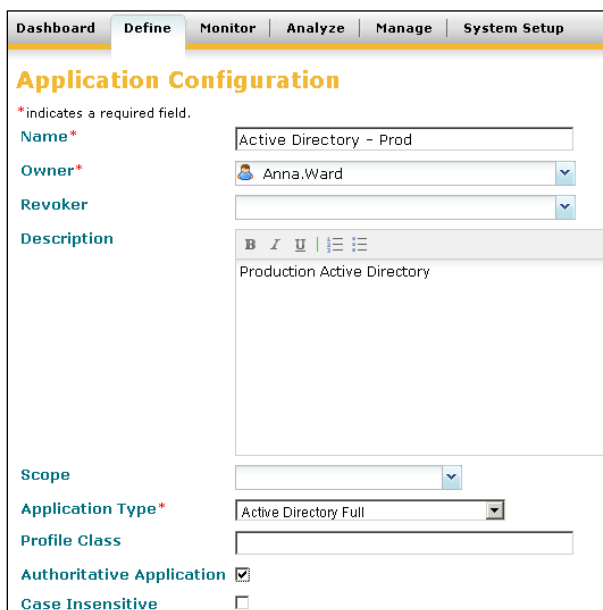
5. Start the Connector Gateway service. The service name is **IdentityIQConnectorGateway_[SERVICE_NAME_SUFFIX]**.
6. Confirm that the service is running (Status = Started in the services interface).

The Connector Gateway log is found in the log directory under the Connector Gateway's installation directory. For troubleshooting, adjust the logging level by modifying the log4j.properties file in the Connector Gateway installation directory (the log4j.rootLogger entry).

Application Definition

In the IdentityIQ application, configure the Active Directory Application to use the Gateway Connector.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.
3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select **Active Directory Full** as the **Application Type**. This tells IdentityIQ to use the AD Gateway Connector.



Dashboard Define Monitor Analyze Manage System Setup

Application Configuration

* indicates a required field.

Name* Active Directory - Prod

Owner* Anna.Ward

Revoker

Description Production Active Directory

Scope

Application Type* Active Directory Full

Profile Class

Authoritative Application ☒

Case Insensitive ☐

Figure 13: Active Directory Full Connector specification

5. On the **Attributes** tab, specify the connection parameters:
 - **Connector Gateway Host:** Host name or IP Address of the machine where the Connector Gateway resides (for CM 6.0 installations, this is the Host or IP address of the CM machine)
 - **Connector Gateway Port:** Port number selected for the Connector Gateway (for CM 6.0 installations, this is the Port for CM)
 - **MSCS Name:** MSCS name specified in the Connector MSCS definition
 - **Username:** DN of the Connector Administrator (defined during configuration of Connector)
 - **Password:** Connector Administrator's password (defined during configuration of Connector)

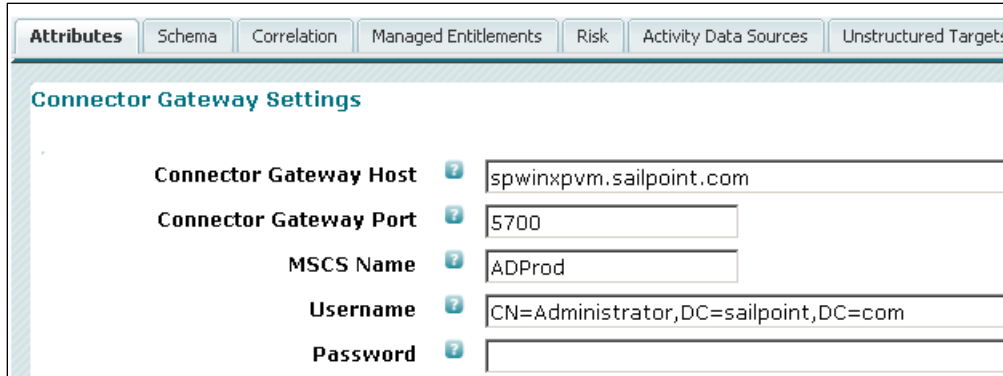


Figure 14: Gateway Connector settings

When encryption is implemented, the Encryption type (Triple DES or DES) and full path of the Encryption file must also be specified on the **Attributes** tab. This encryption specification applies to data encryption between IdentityIQ and the Connector Manager. The Provisioning Engine Guide that is included in the IdentityIQ product documentation (found in the [IdentityIQ Installation Directory]\doc\pdf directory) explains the process for setting up this encryption in the chapter entitled “Enabling Secured Communications Between IdentityIQ and Connector Manager.”

6. Click **Test Connection** (at the bottom of the window) to verify whether IdentityIQ can connect to the Active Directory instance with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successfully made. The message contains information about the connection error if the connection test fails.

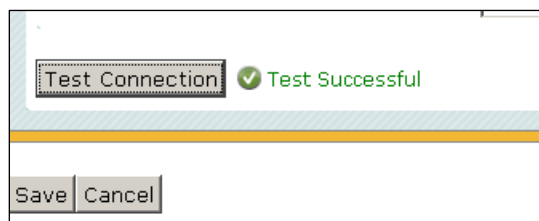


Figure 15: Successful Test Connection

7. Click **Save** to save the application definition.

Specifying the Active Directory Full connector automatically creates a Provisioning Policy for account creation that includes the fields required by Active Directory. It can be modified to set up default values for these fields or to add additional fields. To view or modify the Provisioning Policy, click the **Provisioning Policies** tab and click the policy name link (named **account**, by default).

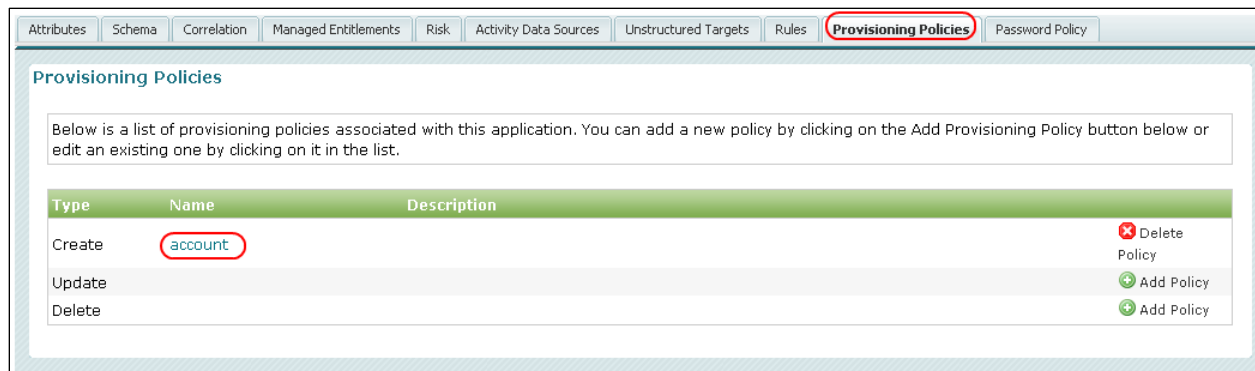


Figure 16: Provisioning Policies List

To view or change the details related to a specific field, click the field name in the left pane to display its current settings in the right pane. To add a field to the provisioning policy, click **Add Field** above the left pane's field list.

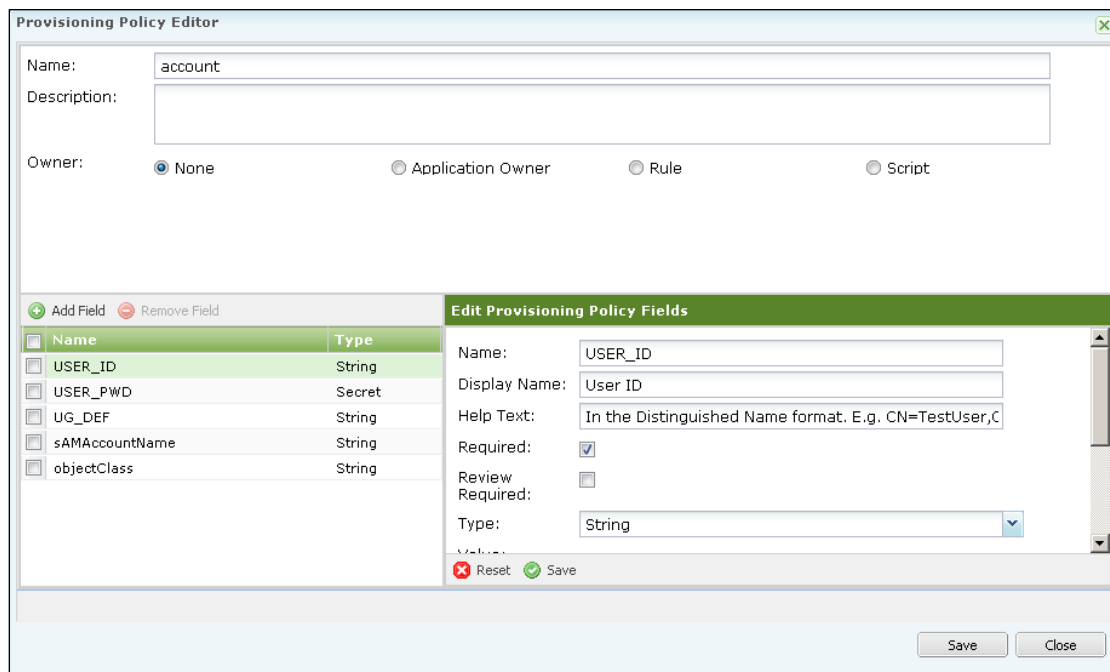


Figure 17: Provisioning Policy Specification

Fields that do not have a default **Value** (or are marked **Review Required**) will be presented to the requesting user when a new account is requested. Note that the fields must be sent to AD in the format in which AD expects them, so either the user will have to enter the data as required or a rule or script will need to be written in the provisioning policy to make the user's entry conform to AD's requirements. This includes password complexity requirements. The **Help Text** attribute can be used to inform the user of the requirements.

Troubleshooting

If "Test Connection" fails or if a requested provisioning action is not completed, some diagnostic activities will be required to determine the problem's cause.

The first diagnostic resource to examine is the log files. The Connector Gateway's log file is stored in the log directory under the Connector Gateway's installation directory ([cgHome]/log/connector_gateway.log). The connector manager's log files are found in the log directory under its installation directory ([cmHome]/log/*.log). Connector Manager's logging produces a larger number of small log files; use the timestamp on the files to identify which log files apply to the failing activities.

The cmdetails command (run cmdetails.bat from within the CM's home directory) provides details about the connector manager (version, port, etc.) and any connectors set up within it. Other tools available for managing Connector Manager are documented in the Connector Manager Administrator Guide (downloadable from [Compass: Downloads -> Product Resources -> Product Releases -> Provisioning Engine -> Documentation](#), filename: 5_5_Connector_Manager_Windows_AG.pdf). Similarly, the Connectors' Admin Guides may contain additional monitoring tools specific to each connector.

Custom Attributes

When custom attributes are added to the Active Directory configuration for provisioning, they must be added to the Connector Manager's keywords.txt file. The CN schema object name of the AD attribute must be used in the keywords.txt file. This may *not* match the LDAP display name shown in the AD user interface and the ADSI editor.

For example, the display name (used in the AD UI) for Extension Attribute 1 is *extensionAttribute1* while the CN is *ms-Exch-Extension-Attribute-1*. The entry for this attribute in the keywords.txt file must specify *ms-Exch-Extension-Attribute-1*, not *extensionAttribute1*.

The entries for the extension attributes in the keywords.txt file should look something like this:

extensionAttribute1	ms-Exch-Extension-Attribute-1	-1	1,2,26	None
extensionAttribute2	ms-Exch-Extension-Attribute-2	-1	1,2,26	None
extensionAttribute3	ms-Exch-Extension-Attribute-3	-1	1,2,26	None
extensionAttribute4	ms-Exch-Extension-Attribute-4	-1	1,2,26	None
extensionAttribute5	ms-Exch-Extension-Attribute-5	-1	1,2,26	None
extensionAttribute6	ms-Exch-Extension-Attribute-6	-1	1,2,26	None
extensionAttribute7	ms-Exch-Extension-Attribute-7	-1	1,2,26	None
extensionAttribute8	ms-Exch-Extension-Attribute-8	-1	1,2,26	None

The Microsoft website contains a good reference document showing the CN (and other attributes) of the available extension attributes: <http://msdn.microsoft.com/en-us/library/ms980473%28v=exchg.65%29.aspx>

LDAP

LDAP (Lightweight Directory Application Protocol) systems can be connected to IdentityIQ with connectors of all three types: Governance, Gateway, and Direct. Though communication with LDAP systems is the same across the applications, certain attributes of the systems can vary slightly, which requires that their connectors be distinct. Having separate connector types defined for each of these systems out of the box minimizes the customization efforts required to connect with these systems. The available LDAP connectors are listed below, grouped by the LDAP application to which they connect.

LDAP Application	Connector	Connector Style	When to Choose
IBM Tivoli Directory Server	Tivoli – Gateway	Gateway	Used when delta aggregation is desired (detecting account changes and keeping IdentityIQ in sync in real time without requiring bulk re-aggregation). Also used when the BMC ESS product is used concurrently with IdentityIQ so the same connector setup can be used for both.
	Tivoli - Direct	Direct	Used for most connection needs with Tivoli Directory Server; aggregation process can be optimized to match delta aggregation in performance and setup incurs much less overhead than connector gateway infrastructure.
ADAM	ADAM – Gateway	Gateway	Used when delta aggregation is desired (detecting account changes and keeping IdentityIQ in sync in real time without requiring bulk re-aggregation). Also used when the BMC ESS product is used concurrently with IdentityIQ so the same connector setup can be used for both.
	ADAM – Direct	Direct	Used for most connection needs with ADAM; aggregation process can be optimized to match delta aggregation in performance and setup incurs much less overhead than connector gateway infrastructure.
Novell eDirectory	Novell eDirectory – Direct	Direct	Used for any connection need to the Novell eDirectory System
Sun ONE Directory Server	SunOne – Gateway	Gateway	Used when delta aggregation is desired (detecting account changes and keeping IdentityIQ in sync in real time without requiring bulk re-aggregation). Also used when the BMC ESS product is used concurrently with IdentityIQ so the same connector setup can be used for both.
	SunOne – Direct	Direct	Used for most connection needs with Sun ONE; aggregation process can be optimized to match delta aggregation in performance and setup incurs much less overhead than connector gateway infrastructure.
Oracle Internet Directory	OID – Direct	Direct	Used for any connection need to the Oracle Internet Directory system
Other LDAP Systems	LDAP	Governance	Generic read-only connector for LDAP systems
	OpenLDAP – Direct	Direct	Generic read-write connector for LDAP systems that follow the Open LDAP standard

LDAP Governance Connectors

At one time, IdentityIQ included several governance connectors to communicate with various LDAP systems. Now, however, most of those have been migrated to Direct connectors, enabling writing functionality as well as reading.

One generic LDAP governance connector remains for installations wishing to connect in read-only mode to an LDAP system. When the LDAP governance connector is used, note that the account and group schemas may need to be altered from the defaults to match the different naming conventions for some attributes. Use of the LDAP Governance connector is expected to be the exception case, rather than the norm, since the Direct connectors can also be used for read-only connectivity and are preconfigured to match the account and group schemas required by each system. To forcibly turn off writing capabilities for a Direct connector, alter its FeaturesString value to include only the options: “Authenticate, Manager Lookup, Search.”

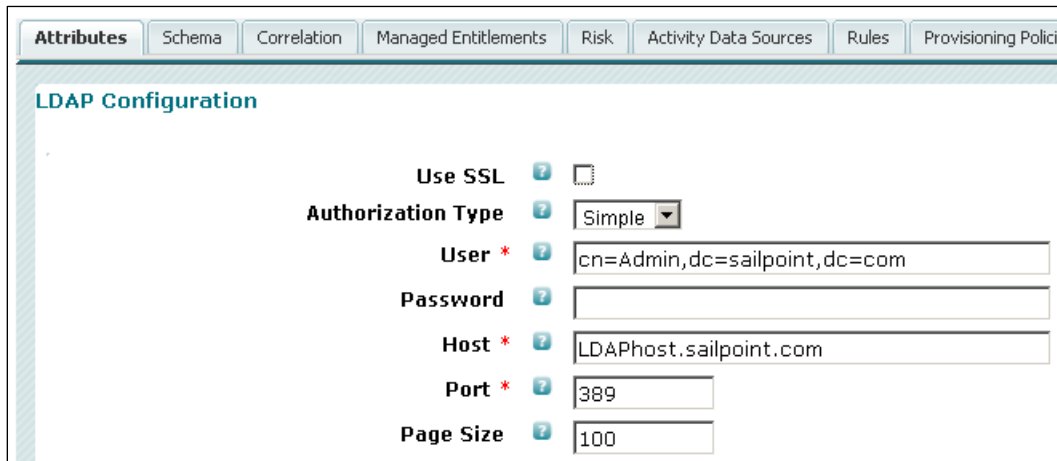
Configure the LDAP application to use a Governance Connector following these simple steps.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.
3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select **LDAP** as the **Application Type**. The required connection parameters are then displayed in the **Attributes** tab.

Figure 18: LDAP Application Configuration

5. On the **Attributes** tab, specify the connection configuration attributes:
 - **Use SSL:** Select if the connection between the IdentityIQ host and the LDAP host uses SSL (appropriate client and server certificates will be required)
 - **Authorization Type:** Simple or None; choosing None bypasses authorization and signs on as an anonymous user (which may not be permitted by your LDAP setup); choosing Simple authenticates with the User and Password provided in the next parameters

- **User:** DN for the user's account used in making the connection to the LDAP system; required
- **Password:** the password for the connection user account
- **Host:** the hostname or IP Address of the LDAP instance's host machine; required
- **Port:** the TCP/IP Port on which the LDAP instance is listening; required (default is 389 for non-SSL or 636 for SSL)



LDAP Configuration

Use SSL ☐

Authorization Type

User *

Password

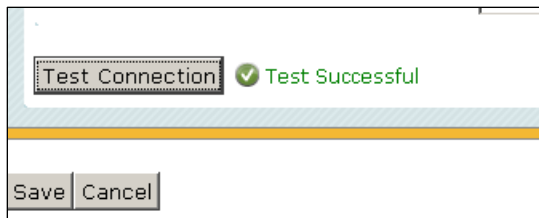
Host *

Port *

Page Size

Figure 19: LDAP Configuration Parameters

6. Click **Test Connection** (at the bottom of the page) to verify whether IdentityIQ can connect to the Active Directory instance with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successful or not.

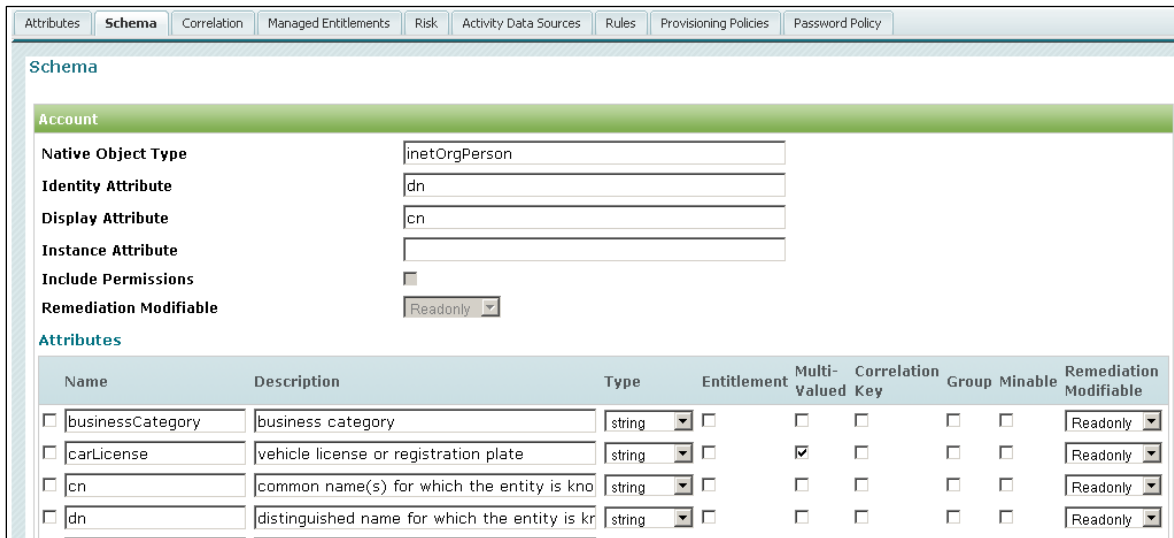


Test Connection ☒ Test Successful

Save Cancel

Figure 20: Successful Test Connection

- If necessary, alter the schema attributes on the **Schema** tab.



Name	Description	Type	Entitlement	Multi-Valued	Key	Correlation	Group	Minable	Remediation Modifiable
<input type="checkbox"/> businessCategory	business category	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> carLicense	vehicle license or registration plate	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> cn	common name(s) for which the entity is known	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> dn	distinguished name for which the entity is known	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly

Figure 21: Application Schema

- Click **Save** to save the application definition.

LDAP Direct Connectors

All of the LDAP Direct Connectors establish their connections between IdentityIQ and the LDAP system identically. However, since different LDAP systems vary in how they manage certain tasks and in how they name certain attributes, their differences are managed through the application definition specified for each connector in the connector registry. In essence, each of the separate connectors predefines the customizations required to connect with each system.

NOTE: Only customers who have licensed IdentityIQ's Provisioning Engine component are permitted to use the provisioning capabilities of these connectors. They can be configured and used for read-only activities by any customer.

Viewing the Connector Configurations

The configurations for each connector type can be viewed, and modified if necessary, through the IdentityIQ Debug pages. Select **Configuration** in the Objects list and click **List**.

Dashboard | **Define** | **Monitor** | **Analyze** | **Manage** | **System Setup** | **Debug**

Debug Pages

Select Debug Operation

System Configuration | Identity Configuration | Link Configuration | UI Configuration

View Configuration Caches | Reset Configuration Caches

Configuration

Run Approval Library

Select **ConnectorRegistry** to view the XML for all the connectors.

Dashboard | **Define** | **Monitor** | **Analyze** | **Manage** | **System Setup** | **Debug**

Debug Pages

Configuration Objects

Name

ActivityCollectorConfigPageRegistry

ActivityCollectorRegistry

ConnectorRegistry

IdentitySelectorConfiguration

SystemConfiguration

The customized schemas, provisioning policies, attributes, etc. are all specified within each application definition.

Dashboard | **Define** | **Monitor** | **Analyze** | **Manage** | **System Setup** | **Debug**

Debug Pages

Edit Configuration:ConnectorRegistry

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Configuration PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Configuration created="1317070965870" id="40288f0132a78bb90132a78c0c6e0091" modified="1333380941224" name="ConnectorRe">
  <Attributes>
    <Map>
      <entry key="applicationTemplates">
        <value>
          <List>
            <Application connector="sailpoint.connector.ADLDAPConnector" featuresString="AUTHENTICATE, MANAGER_LOOKUP, SEARCH">
              <Attributes>
                <Map>
                  <entry key="formPath" value="ldapAttributesForm.xhtml"/>
                </Map>
              </Attributes>
            </Application>
          </List>
        </value>
      </entry>
    </Map>
    <Schema displayAttribute="sAMAccountName" identityAttribute="distinguishedName" nativeObjectType="User" objectType="User">
      <AttributeDefinition name="businessCategory" type="string">
        <Description>business category</Description>
      </AttributeDefinition>
    </Schema>
  </Attributes>
</Configuration>
```


LDAP Connector Customizations

All of the LDAP connectors include specification of the Account and Group schemas as required by the associated LDAP system. They also include an account creation provisioning policy with the required fields for creating an account in that system. Additional customizations implemented for each of these LDAP systems are summarized in this table:

LDAP Application	Connector Customizations
ADAM	<ul style="list-style-type: none"> Features: Provisioning, Enable, Password, Authenticate, Manager Lookup, Search EnableAccountAttr=True allows account to be enabled during creation (by default, ADAM creates in a non-activated state and must be explicitly activated to be used) Restore account by setting msDS-UserAccountDisabled to False Revoke Account by setting msDS-UserAccountDisabled to True
Novell eDirectory	<ul style="list-style-type: none"> Features: Provisioning, Enable, Unlock, Authenticate, Manager Lookup, Search Iterate Mode set to virtual list view instead of paged results Lock attribute: lockedByIntruder = True Restore account by setting loginDisabled to null Revoke Account by setting loginDisabled to True Unlock account by setting lockedByIntruder to null
Tivoli Directory Server	<ul style="list-style-type: none"> Features: Provisioning, Password, Authenticate, Manager Lookup, Search
Oracle Internet Directory	<ul style="list-style-type: none"> Features: Provisioning, Enable, Unlock, Password, Authenticate, Manager Lookup, Search Lock attribute: pwdaccountlockedtime is non-null Restore account by setting orclIsEnabled to null Revoke Account by setting orclIsEnabled to DISABLED Unlock account by setting orclpwdaccountunlock to 1 (OID then automatically resets pwdaccountlockedtime to null) Plan Initializer Script: For account creation requests, appends "orclUser" and "orclUserV2" to the value field of associated attribute requests with name "objectclass"
Sun ONE Directory Server	<ul style="list-style-type: none"> Features: Provisioning, Enable, Unlock, Password, Authenticate, Manager Lookup, Search Lock attribute: passwordretrycount = 3 Restore account by setting nsRoleDN to "cn=nsManagedDisabledRole,dc=Naming Context" Revoke account by setting nsRoleDN to null if it is currently set to "cn=nsManagedDisabledRole,dc=Naming Context" (active account value) NOTE: For both Restore and Revoke, "Naming Context" must be changed in the Connector Registry to a value appropriate for the installation's DN (e.g. cn=nsManagedDisabledRole,dc=sailpoint,dc=com) Unlock account by setting passwordretrycount to null if it is currently 3

NOTE: Once a connector has been assigned to an application and the application definition has been saved, changes to the connector for that application instance must be made in the Application xml instead of in the

connector registry. Changes applied to the connector registry will affect any applications that are configured to use that connector *after* the change is made, but they will not affect existing applications configured with that connector. The application xml can be viewed from the IdentityIQ Debug pages: Select **Application** from the Object list and click **List**. Then select the application by name from the **Application Object** list to view and edit its XML representation.

Debug Pages

Select Debug Operation

System Configuration
Identity Configuration
Link Configuration
UI Configuration

View Configuration Caches
Reset Configuration Caches

Application
List
Edit

Run
Approval Library

Debug Pages

Application Objects

Name

- Active Directory - Production
- AD Production GW
- ADAM
- ADAM Direct
- Enterprise Directory - CONTRACTORS
- Enterprise Directory - EMPLOYEES

Debug Pages

Edit Application:ADAM Direct

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Application PUBLIC "sailpoint.dtd" "sailpoint.dtd">
<Application connector="sailpoint.connector.LDAPConnector" created="1334252935835" featuresString="AUTHENTICATE, PROVISIONING"
  <AccountCorrelationConfig>
    <Reference class="sailpoint.object.CorrelationConfig" id="4028833636890f860136a80da5ab057d" name="ADAM Direct"/>
  </AccountCorrelationConfig>
  <Attributes>
    <Map>
      <entry key="acctAggregationEnd">
        <value>
          <Date>1334599437343</Date>
        </value>
      </entry>
      <entry key="acctAggregationStart">
        <value>
          <Date>1334599298296</Date>
        </value>
      </entry>
    </Map>
  </Attributes>
</Application>

```

Available Customization Parameters

Most of the time, the application attributes and schemas will not need to be changed for these applications when the correct connector type is selected; the very reason for the existence of multiple connector types is to pre-configure the required customizations for each LDAP system.

However, some installations may have a custom configuration of their LDAP system that requires changes to be applied to the connector. The table below lists the attributes that can be specified within an application definition to affect the connector's operations. These are specified within the Attribute map section of the application definition. **NOTE:** It is *strongly* advised that these attributes only be changed from the defaults configured for the connector (including adding new attributes) when the attributes' usage is fully understood by

the implementer and the change has been determined to be necessary to address a specific need of the installation.

Attribute	Purpose
groupEntitlementAttr	Name of the attribute that represents the entitlement
groupMemberAttr	Name of group membership attribute (default is uniqueMember if not specified)
revokeAttr	Attribute modified to revoke/disable an account
revokeVal	revokeAttr set to this value to revoke/disable account
restoreAction	Valid Values: Add, Replace Determines how restoration (account enable) is processed If action = Add or action = Replace with a restoreVal provided, restoreAttr set to restoreVal If action = Replace w/no restoreVal, restoreAttr set to null If action omitted or set to anything other than Add or Replace, restoreAttr set to null if it currently matches specified restoreVal
restoreAttr	Attribute modified to restore/re-enable an account
restoreVal	Value applied in restore action
unlockAction	Valid Values: Add, Replace Determines how account unlock is processed If action = Add or action = Replace with a restoreVal provided, restoreAttr set to restoreVal If action = Replace w/no restoreVal, restoreAttr set to null If action omitted or set to anything other than Add or Replace, restoreAttr set to null if it currently matches specified restoreVal
unlockAttr	Attribute modified to unlock an account
unlockVal	Value applied in unlock action
passwordAttr	Attribute that stores the user's password
enableAccountAttr	Boolean that determines whether an account restore (enable) request is required during processing of a "create" provisioning request
lockAttr	Attribute that indicates account is locked
lockVal	Value of lockAttr that signifies locked status (if omitted, any non-null value in lockAttr signifies locked)
useSSL	Sets security protocol to SSL; Required for any password activities (password reset, account creation with password, etc.)
authorizationType	Valid Values: None, Simple Authorization type to use for connection
searchDN searchDNs searchCountLimit searchTimeLimit searchDeferLink searchReturningObj	Attributes used to set Java searchControls class attributes to other than the default values

groupMemberSearchDN groupMemberSearchScope groupMemberFilterString	Attributes used to override group search defaults
referral	Valid values: follow, ignore, throw LDAP DirContext Referral value Default: follow
groupMemberAttribute	Overrides default group membership attribute name Default: uniqueMember
iterateModeOverride	Valid Values: VIRTUAL_LIST_VIEW, PAGED_RESULTS Overrides the default iteration mode for iterating over objects on the host
disablePooling	Boolean to determine whether connection pooling is disabled or not
disableSort	Boolean to determine whether sort is disabled or not on search (used in setting up iterator)
sortAttribute	Identifies sort attribute to use in setting up iterator (CN is default)
sortCritical	Boolean to determine whether to use critical or noncritical sort control (critical means sort must be honored by LDAP server or must refuse to perform the search; noncritical means sort may be honored or not)
authSearchAttributes	Valid Values: list of attribute names Identifies list of attributes to use in pass through authentication search (if omitted, default list is used) Specified as list; example: <pre> <entry key="authSearchAttributes"> <value> <List> <String>cn</String> <String>uid</String> <String>mail</String> </List> </value> </pre> Can also be specified through application configuration UI
ldapContextFactory	LDAP context factory (default is com.sun.jndi.ldap.LdapCtxFactory if not provided)
expiredPasswordErrorMessages	Valid Values: CSV list of expired password messages Authentication exception message is compared to this list to determine if it matches one of these and, if so, marks the exception as a password expiration exception Default strings are "password expired" and "expired password" if this override is not specified
dnSearchFilter	Override filter string used in lookup of DN from server to get actual stored value of DN (passed to LDAP DirContext Search to get NameInNamespace); Default filterString is "(&(cn=*))" and is believed to work in all cases; this attribute is only here to allow override if that default string does not work for a specific installation
disableDNSearch	Boolean to disable lookup of DN from server to get actual stored

	value for DN; this attribute is specified with the schema type so it can be activated for each schema independently (group.disableDNSearch or account.disableDNSearch)
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

NOTE: All Boolean attributes default to False when the attribute is omitted from the Connector Registry's application definition XML.

If needed, changes can be made to the account and group schemas for the application. Edit the XML or adjust the schemas through the UI when configuring the application connection. In most cases, this should not be necessary, since the schemas are customized per connector to match each LDAP application's default schema.

Configuration Steps

Configure the LDAP application to use a Direct Connector following these simple steps.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.
3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select the **Application Type** as shown in the table below. The required connection parameters are then displayed in the **Attributes** tab.

LDAP System	Application Type
ADAM	ADAM – Direct
Novell eDirectory	Novell eDirectory – Direct
Oracle Internet Directory	OID - Direct
SunOne	SunOne - Direct
Tivoli	Tivoli - Direct
Other LDAP systems	OpenLDAP - Direct

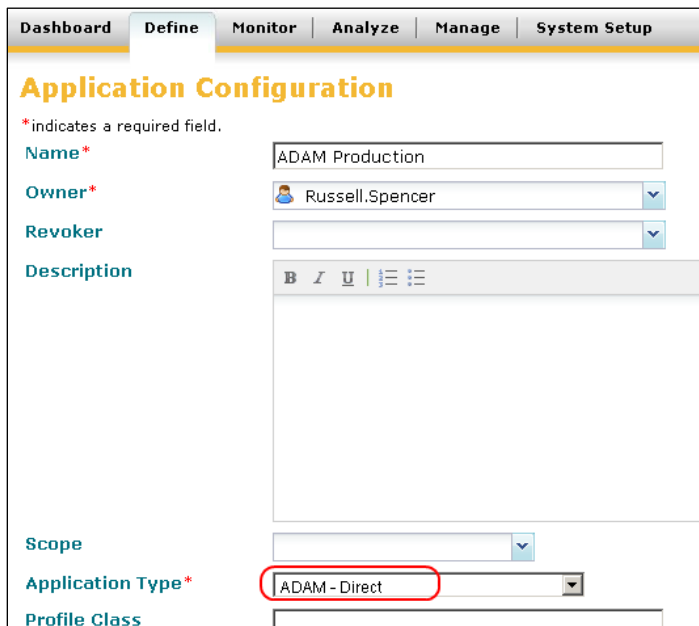


Figure 22: LDAP Application Configuration

5. On the **Attributes** tab, specify the connection configuration attributes:
 - **Use SSL:** Select if the connection between the IdentityIQ host and the LDAP host uses SSL (appropriate client and server certificates will be required)
 - **Authorization Type:** Simple or None; choosing None bypasses authorization and signs on as an anonymous user (which may not be permitted by your LDAP setup); choosing Simple authenticates with the User and Password provided in the next parameters
 - **User:** DN for the user's account used in making the connection to the LDAP system; required
 - **Password:** the password for the connection user account
 - **Host:** the hostname or IP Address of the LDAP instance's host machine; required
 - **Port:** the TCP/IP Port on which the LDAP instance is listening; required (this is the lower of the two sequential port numbers specified during LDAP setup)

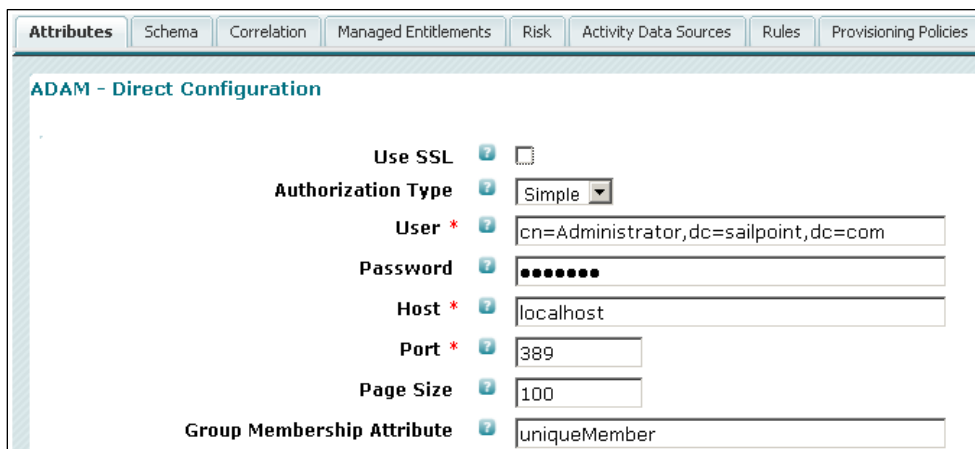


Figure 23: ADAM-Direct Configuration Parameters

6. Click **Test Connection** (at the bottom of the page) to verify whether IdentityIQ can connect to the Active Directory instance with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successful or not. If it is unsuccessful, the message indicates the error that must be corrected.

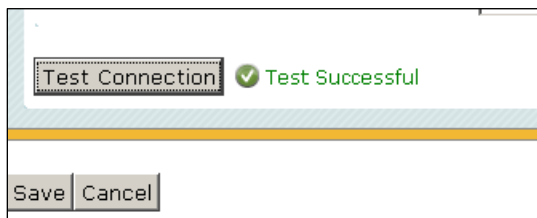


Figure 24: Successful Test Connection

7. On the **Account** tab, alter the **Account Settings Search DN** (and **Search Scope**, if applicable) to match your ADAM instance. This is the DN from which to start a directory search. Other values that can be set on this tab are:
 - **Search Scope:** scope of search within LDAP directory tree (options: Base, One Level, Subtree)
 - **Group Member Search DN:** specifies DN that should be used when searching for a user's groups. If this is not defined the Search DN is used.

- **Iterate Search Filter:** LDAP-server-side filter to scope down the object set being returned
- **Filter String:** SailPoint filter string that allows IdentityIQ-server-side filtering of objects (not generally used with LDAP applications since LDAP-server-side filter is available)

Separate values for these search settings can be defined on the Group tab for group searches if appropriate; if not specified, the Account search parameters will be applied to Groups as well.

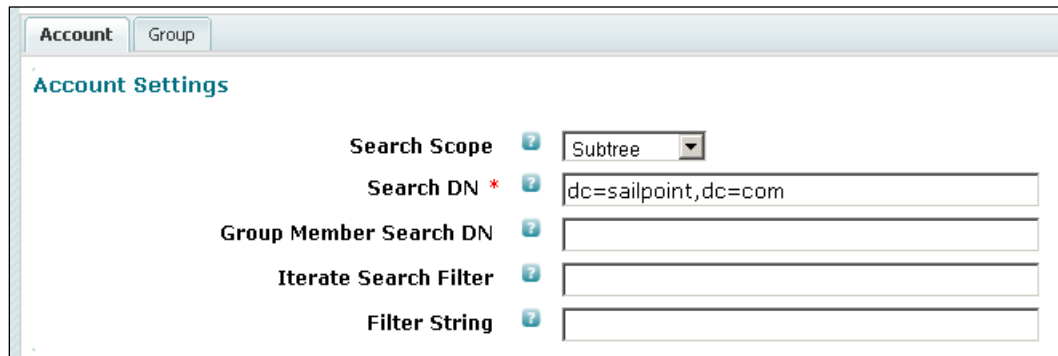
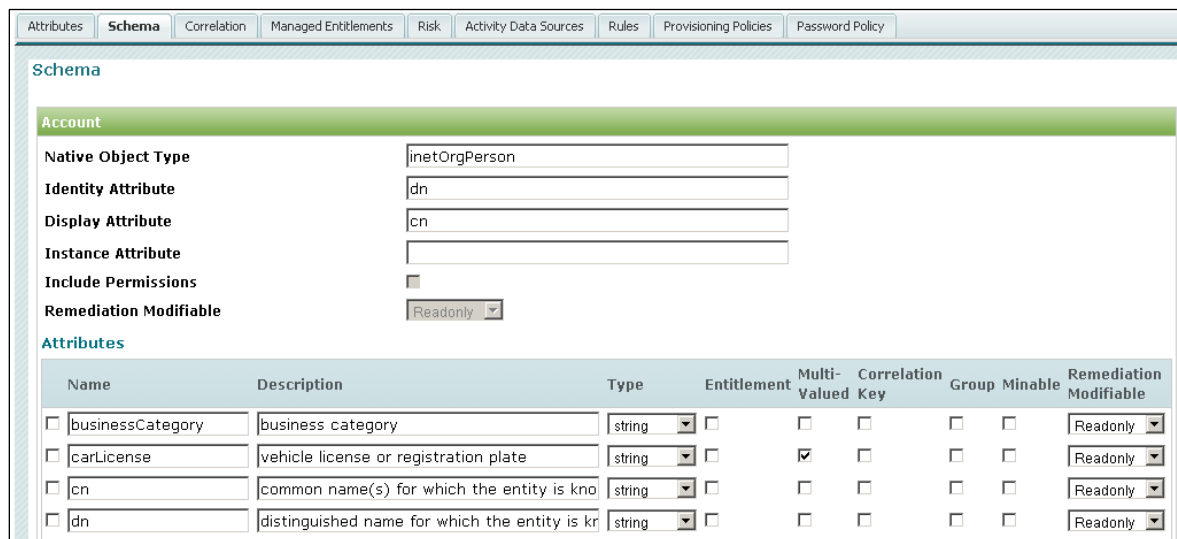


Figure 25: Account Settings Tab

8. If necessary, alter the schema attributes on the **Schema** tab.



Name	Description	Type	Entitlement	Multi-Valued	Correlation Key	Group	Minable	Remediation Modifiable
<input type="checkbox"/> businessCategory	business category	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ReadOnly
<input type="checkbox"/> carLicense	vehicle license or registration plate	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ReadOnly
<input type="checkbox"/> cn	common name(s) for which the entity is known	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ReadOnly
<input type="checkbox"/> dn	distinguished name for which the entity is known	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ReadOnly

Figure 26: Application Schema

9. Click **Save** to save the application definition.

LDAP Gateway Connectors

Gateway connectors exist for SunOne, Tivoli, and ADAM. As noted in the table at the beginning of the LDAP chapter, these are typically used when delta aggregation is being used to keep IdentityIQ in sync with the LDAP system in real-time without requiring a bulk re-aggregation. Installations not requiring real-time updates, however, may find that the relative simplicity of the direct connectors' installation/configuration process may make them preferable to the Gateway Connectors.

Setting up these LDAP systems with a read-write Gateway Connector requires these steps to be done in the order specified here.

1. Complete Pre-installation Activities
2. Install the Connector Manager
3. Install the Connector
4. Install the Connector Gateway
5. Specify the Active Directory Application Configuration in IdentityIQ

Pre-Installation Activities

The required pre-installation activities depend upon the operating system on which the connector manager and connector will be installed. For ADAM, this will be a Windows platform. For SunOne, it will be Windows or Solaris OS and for Tivoli, it can be Windows or AIX.

Refer to the Connector Manager Installation Guide for details on the required pre-installation activities to prepare the environment for a successful installation. Use the table below to determine which installation guide version to use. These guides are provided with the installation files.

OS	CM Version	Installation Guide Filename
Windows	5.2 or 5.5	5_2_Connector_Manager_Windows_IG.pdf
	6.0	6_0_ConnectorManager_Windows_Install_Guide.pdf
Any UNIX OS	5.2 or 5.5	5_2_Connector_Manager_UNIX_IG.pdf
	6.0	6_0_ConnectorManager_UNIX_Install_Guide.pdf

These activities include:

- CM Owner/ installation account selection
- TCP/IP port selection
- Encryption plan
- JRE installation (UNIX only)
- UNIX Kernel requirements (UNIX only)
- C++ Runtime installation (Windows only)

Connector Manager Installation

As with the pre-installation activities, the procedures for installing Connector Manager depend on the platform on which it is being installed and the CM version being installed. The ADAM Connector Manager must be installed on a Windows machine. The Other two may be installed on Windows, AIX, or Solaris.

Install the Connector Manager and any patches required as indicated in the appropriate installation guide.

The table below shows the installation files and patch files required for each of the most recent CM versions on each of the OS platforms.

IdentityIQ Versions	CM Version	OS	Installation File	Patch File
---------------------	------------	----	-------------------	------------

5.2	5.2	Windows	ConnectorManager-Windows-xxbit-5.2p1.zip	None
		AIX	ConnectorManager-AIX-5.2p1.tar	None
		Solaris	ConnectorManager-Solaris-sparc-5.2p1.tar ConnectorManager-Solaris-x86-5.2p1.tar	None
5.5GA – 5.5p2	5.5	Windows	ConnectorManager-Windows-xxbit-5.2p1.zip	ConnectorManager-Windows-xxbit-5.5.00-patch.zip
		AIX	ConnectorManager-AIX-5.2p1.tar	ConnectorManager-AIX-5.5.00-patch.tar
		Solaris	ConnectorManager-Solaris-sparc-5.2p1.tar ConnectorManager-Solaris-x86-5.2p1.tar	ConnectorManager-Solaris-5.5.00-patch.tar (contains binaries for all of the Solaris platforms: Sparc 32-bit and 64-bit and x86 32-bit and 64-bit)
5.5p3+	6.0	Windows	ConnectorManager-Windows-xxbit-6.0.zip	None
		AIX	ConnectorManager-AIX-6.0.zip	None
		Solaris	ConnectorManager-Solaris-sparc-6.0.zip ConnectorManager-Solaris-x86-6.0.zip	None

Installation instructions are found in these documents:

Installation/Patch File	Instruction Document
ConnectorManager-Windows-xxbit-5.2p1.zip	5_2_ConnectorManager_Windows_IG.pdf
ConnectorManager-AIX-5.2p1.tar	5_2_ConnectorManager_Unix_IG.pdf
ConnectorManager-Solaris-sparc-5.2p1.tar	
ConnectorManager-Solaris-x86-5.2p1.tar	
ConnectorManager-Windows-xxbit-5.5.00-patch.zip	5_5_ConnectorManager_Install_Guide.pdf
ConnectorManager-AIX-5.5.00-patch.tar	
ConnectorManager-Solaris-5.5.00-patch.tar	
ConnectorManager-Windows-xxbit-6.0.zip	6_0_ConnectorManager_Windows_Install_Guide.pdf
ConnectorManager-AIX-6.0.zip	6_0_ConnectorManager_UNIX_Install_Guide.pdf
ConnectorManager-Solaris-sparc-6.0.zip ConnectorManager-Solaris-x86-6.0.zip	

Connector Installation

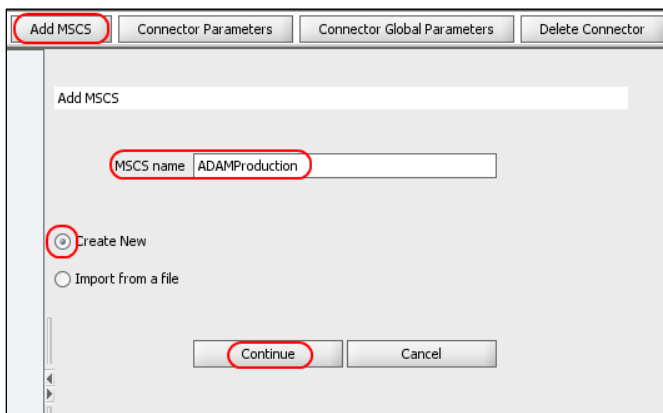
The Connector for LDAP directories can manage the LDAP objects in the entire LDAP Directory information tree. This connector is the same for all LDAP systems and can be installed within any of the CMs to which it applies (Windows, Solaris, AIX). Complete these steps to install and configure the Connector. The specific instructions below (and the related screen shots) relate to the CM for Windows.

1. Install the Connector for LDAP.

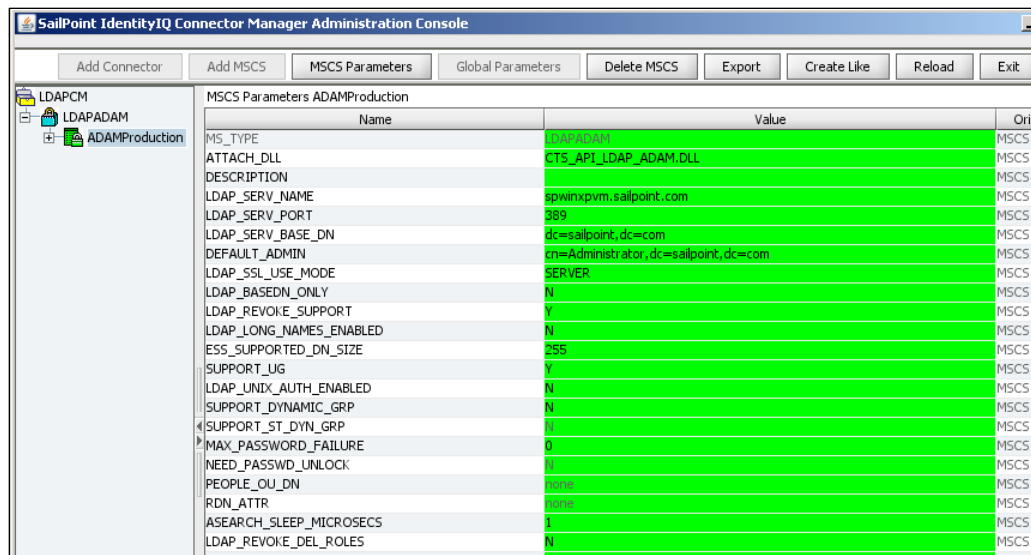
- Obtain the installation files from your SailPoint representative. Select the most recent version. At the time of this writing, the most recent version is in the file Connector-LDAP-5.2.rar.
 - Instructions for installing the connector (and its earliest patches) are in the file: 5_5_Connectors_Install_Guide.pdf.
 - Apply the service pack/patches as described in 5_5_Connectors_Install_Guide.pdf.
2. Configure the Connector's MSCS parameters through the Connector Manager Configuration Console.

NOTE: Screen shots included in this step are from an LDAPADAM connector within a Windows Connector Manager.

- Launch the Connector Manager Configuration Console:
 - In Windows, click **Start -> SailPoint -> Connector Manager -> [CM instance name] > CM Administration Console.**
 - In UNIX, enter **CMAdmin.sh** at a UNIX shell prompt.
NOTE: JRE version 1.6.0_18 or later must be installed for this command to run.
- Select the Connector (listed by the connector's MSCS Type: **LDAPADAM** for ADAM, **LDAPIP** for SunOne, or **LDAPT16.0** for Tivoli) in the tree in the left pane and click **Add MSCS**.
- Specify an **MSCS Name** to use for the managed system. This name is assigned here and entered in IdentityIQ to indicate the managed system to which the connection applies; this name is case-sensitive. Select **Create New** and click **Continue**.



- Enter the values required for connecting to the LDAP instance. The required parameters for each of the 3 LDAP system types are listed in the 5_5_Connectors_Install_Guide.pdf.



3. Start the CM Service. In Windows, this can be done through the Services UI. In UNIX, start it with the command `start-cm.sh`.

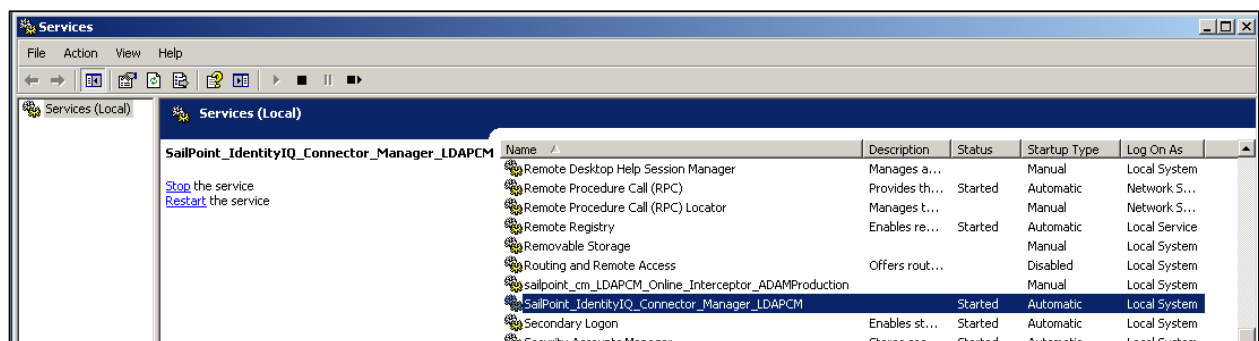


Figure 27: Windows Services UI

Connector Gateway Installation

The Connector Gateway is the intermediary between IdentityIQ and the ConnectorManager. Beginning with CM version 6.0, which pairs with IdentityIQ versions 5.5p3 and later, the Connector Gateway is included within CM and is no longer installed as a separate component. Installations upgrading from CM 5.2 or 5.5 should follow the instructions in the 6.0 Connector Manager Installation guide to remove the external Connector Gateway from the infrastructure:

- Stop the Connector Gateway service.
- Change the IdentityIQ application properties to use the Host and Port details for Connector Manager (instead of the Connector Gateway Host and Port).

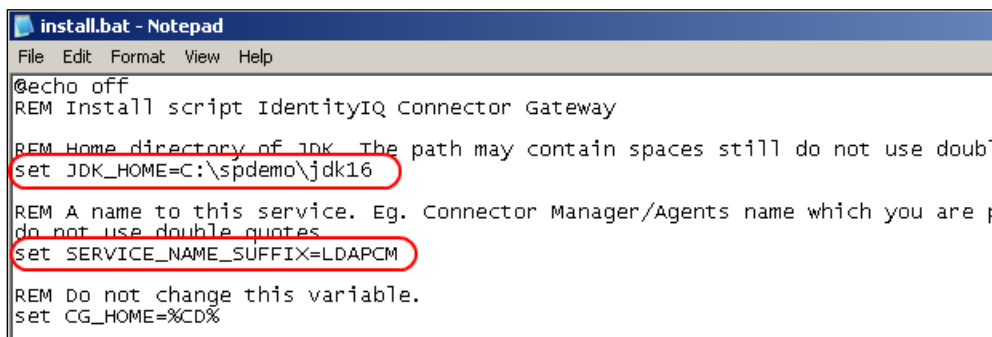
The rest of the Connector Gateway installation instructions included here apply only to installations running CM 5.5 or 5.2, which must install the separate Connector Gateway. It can be installed on any machine with network connectivity to both IdentityIQ and the ConnectorManager, but it is strongly recommended that it be installed on the same machine as Connector Manager where possible. One Connector Gateway is required per ConnectorManager.

The Host name (or IP Address) and Port for the Connector Manager must be known to complete the Connector Gateway setup.

NOTE: Instructions for installing and configuring the Connector Gateway are also included in the README_WIN.TXT and README_UNIX.TXT files within the Connector Gateway's zip file. Other general notes, as well as uninstallation instructions, can also be found in those files.

Complete these steps to install and configure the Connector Gateway on a Windows machine:

1. Locate and unzip the Connector Gateway zip file into the directory location where the Connector Gateway will reside. Connector Gateway is part of the IdentityIQ package; install the Connector Gateway version that matches the IdentityIQ version in use. It can be installed in any directory location.
2. Navigate to the Connector Gateway home directory and edit the **install.bat** file to set the JDK directory and the service name:
 - **set JDK_HOME=[JDK directory path]** (JDK version 1.5 or 1.6 is required. Download and install this if no JDK is present on the machine.)
 - **set SERVICE_NAME_SUFFIX=[service name]** (This assigns the specified name to this Connector Gateway service instance; different Connector Gateway services running on the same machine can be assigned different names. For ease of support, it is recommended that the service name suffix matches, or in some way reflects, the associated CM instance.)



```

install.bat - Notepad
File Edit Format View Help
@echo off
REM Install script IdentityIQ Connector Gateway
REM Home directory of JDK. The path may contain spaces still do not use double
set JDK_HOME=C:\spdemo\jdk16
REM A name to this service. Eg. Connector Manager/Agents name which you are p
do not use double quotes
set SERVICE_NAME_SUFFIX=LDAPCM
REM Do not change this variable.
set CG_HOME=%CD%
  
```

Figure 28: Connector Gateway Service Settings

3. Run **install.bat** to create the service.
4. Edit the **init.xml** file in the Connector Gateway's home directory to set the hostname and port of the Connector Manager and the Port for the Connector Gateway. The Connector Gateway's port can be any available port number you choose. This number will be referenced in the IdentityIQ Application Configuration.

```

<ConnectorGateway>

  <SM>
    <!-- Connector Manager/Agents Hostname or IP Address-->
    <hostname>spwinxpvm.sailpoint.com</hostname>

    <!--Connector Manager/Agents port number-->
    <port>2472</port>

    <!-- Use "AS400" for AS400 system and "MAINFRAME" for Mainframes and
    leave empty in all other cases.-->
  
```

```

        <platform></platform>
    </SM>

    <Server>
        <!-- Connector Gateway port number-->
        <port>5720</port>

        <!-- Delay(in seconds) between two retry attempts while connecting to
Connector Manager/Agents-->
        <sm_connect_retry>3</sm_connect_retry>

    </Server>

</ConnectorGateway>

```

5. Start the Connector Gateway service. The service name is **IdentityIQConnectorGateway_[SERVICE_NAME_SUFFIX]**.
6. Confirm that the service is running (Status = Started in the services interface).

Complete these steps to install and configure the Connector Gateway on a UNIX machine:

1. Add the bin directory of the JDK to your PATH.
2. Edit the **init.xml** file in the Connector Gateway's home directory to set the hostname and port of the Connector Manager and the Port for the Connector Gateway. The Connector Gateway's port can be any available port number you choose. This number will be referenced in the IdentityIQ Application Configuration.

```

<ConnectorGateway>

    <SM>
        <!-- Connector Manager/Agents Hostname or IP Address-->
        <hostname>192.168.198.129</hostname>

        <!--Connector Manager/Agents port number-->
        <port>2470</port>

        <!-- Use "AS400" for AS400 system and "MAINFRAME" for Mainframes and
leave empty in all other cases.-->
        <platform></platform>
    </SM>

    <Server>
        <!-- Connector Gateway port number-->
        <port>5700</port>

        <!-- Delay(in seconds) between two retry attempts while connecting to
Connector Manager/Agents-->
        <sm_connect_retry>3</sm_connect_retry>

    </Server>

</ConnectorGateway>

```

3. Start the IdentityIQ Connector Gateway by running the command:

```
java -jar ConnectorGateway.jar
```

Application Definition

In the IdentityIQ application, configure the LDAP Application to use the Gateway Connector.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.
3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select **Tivoli – Gateway**, **SunOne – Gateway**, or **ADAM – Gateway** (as appropriate to the LDAP system) as the **Application Type**.

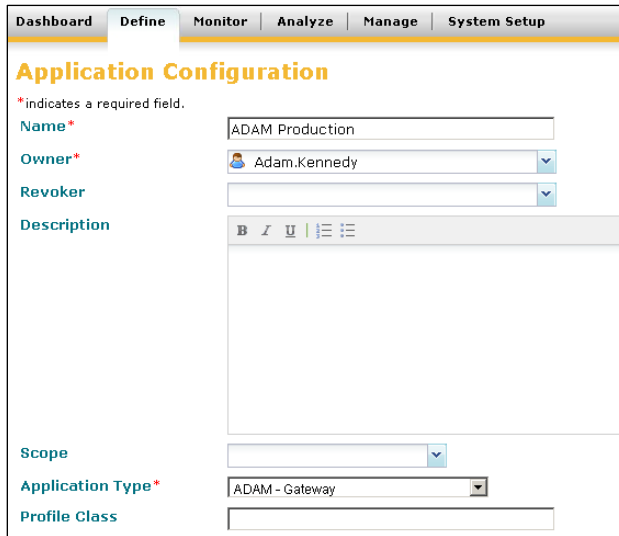


Figure 29: Active Directory Full Connector specification

5. On the **Attributes** tab, specify the connection parameters:
 - **Connector Gateway Host:** Host name or IP Address of the machine where the Connector Gateway resides
 - **Connector Gateway Port:** Port number selected for the Connector Gateway
 - **MSCS Name:** MSCS name specified in the Connector MSCS definition
 - **Username:** DN of the Connector Administrator (defined during configuration of Connector)
 - **Password:** Connector Administrator's password (defined during configuration of Connector)

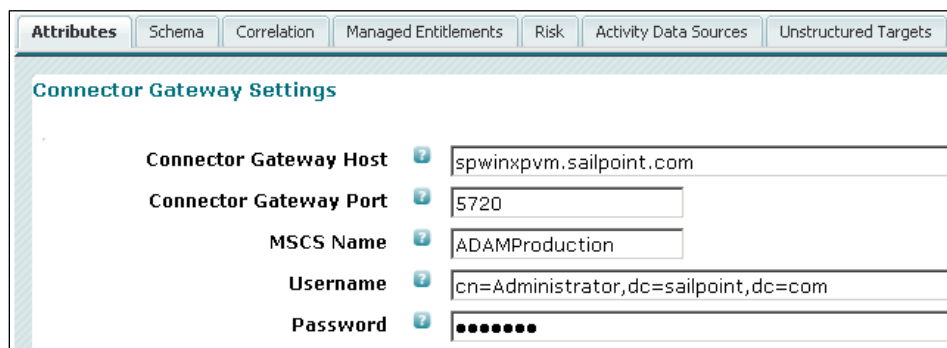


Figure 30: Gateway Connector settings

When encryption is implemented, the Encryption type (Triple DES or DES) and full path of the Encryption file must also be specified here. This encryption specification applies to data encryption between IdentityIQ and the Connector Manager. The Provisioning Engine Guide that is included in the IdentityIQ product documentation explains the process for setting up this encryption in the chapter entitled **Enabling Secured Communications Between IdentityIQ and Connector Manager**.

6. Click **Test Connection** (at the bottom of the window) to verify whether IdentityIQ can connect to the Active Directory instance with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successfully made.

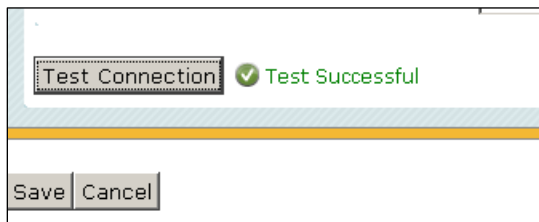


Figure 31: Successful Test Connection

7. Click **Save** to save the application definition.

Specifying the LDAP Gateway connectors automatically creates a Provisioning Policy for account creation that includes the fields required by the specific LDAP system. It can be modified to set up default values for these fields or to add additional fields. To view or modify the Provisioning Policy, click the **Provisioning Policies** tab and click the policy name link (**account**, by default).

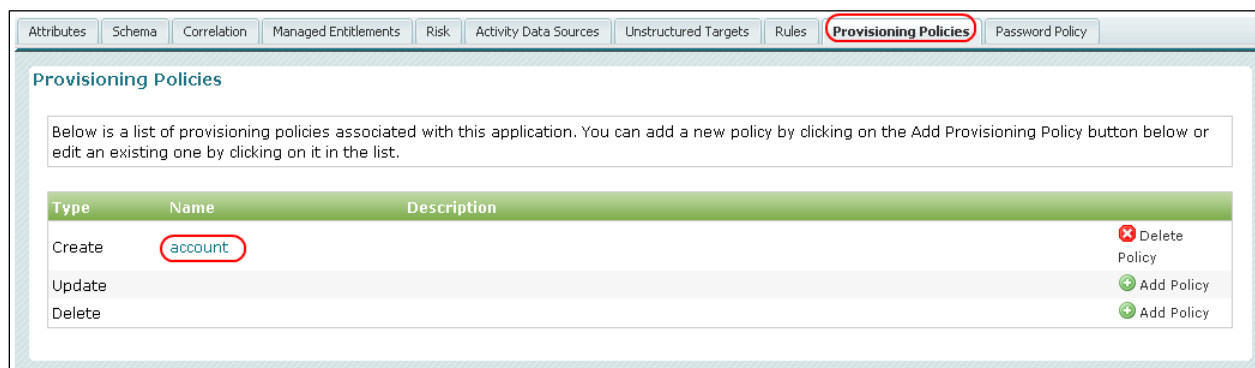
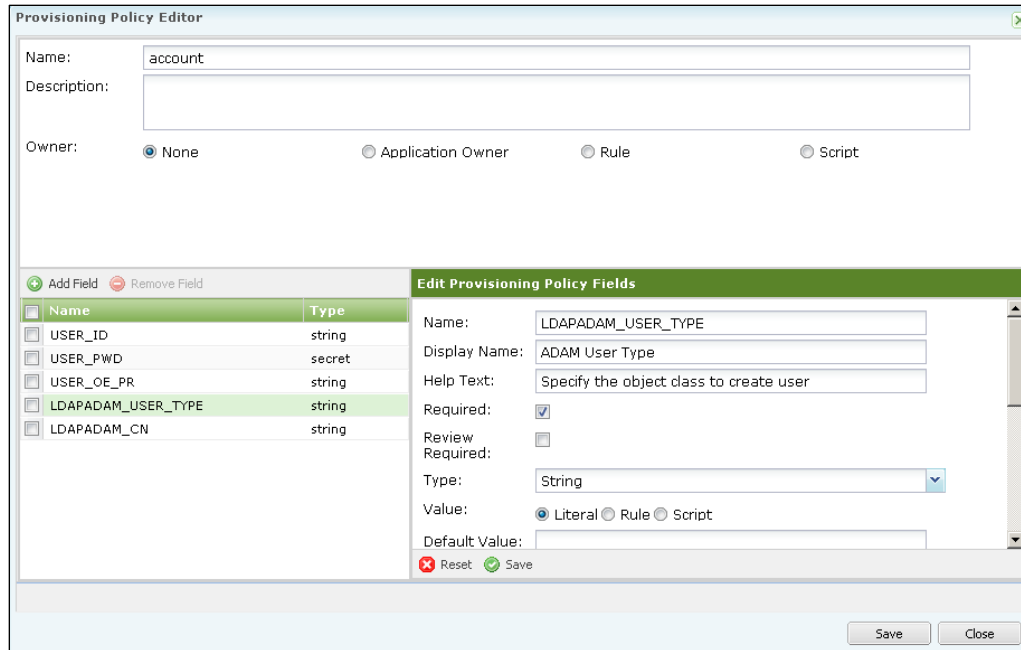


Figure 32: Provisioning Policies List

To view or change the details related to a specific field, click the field name in the left pane to display its current settings in the right pane. To add a field to the provisioning policy, click **Add Field** above the left pane's field list.



Provisioning Policy Editor

Name:

Description:

Owner: ☒ None ☐ Application Owner ☐ Rule ☐ Script

Edit Provisioning Policy Fields

Name	Type
USER_ID	string
USER_PWD	secret
USER_OE_PR	string
LDAPADAM_USER_TYPE	string
LDAPADAM_CN	string

Name:
 Display Name:
 Help Text:
 Required: ☒
 Review Required: ☐
 Type:
 Value: ☒ Literal ☐ Rule ☐ Script
 Default Value:

Figure 33: Provisioning Policy Specification

Fields that do not have a default **Value** (or are marked **Review Required**) will be presented to the requesting user when a new account is requested. Note that the fields must be sent to the LDAP system in the format in which it expects them, so the user will have to enter the data as required or a rule or script will need to be written in the provisioning policy to make the user's entry conform to the system's requirements (including password complexity requirements). The **Help Text** attribute can be used to inform the user of the requirements.

JDBC

The JDBC Connector is a Direct connector that can be used to read data from and write data to a JDBC-enabled database engine. This connector was converted from a Governance connector to a Direct connector in release 5.2p1; prior to that, this connector only had read capabilities. The Direct connector replaces the Governance connector, so only one is available in any given IdentityIQ implementation. Though this Direct Connector provides provisioning capabilities, they are, as always, to be used only by customers who have licensed the product's provisioning engine.

JDBC Direct Connector

Implementing this connector requires specification of the parameters for connecting to the database and the SQL query to retrieve the desired data fields from the database. Information on how rows should be merged, if applicable, can also be specified.

Configure an application to use the JDBC Connector following these simple steps.

1. Click **Define** -> **Applications**.
2. Click **New Application...** to open an **Application Configuration** window.
3. Enter the application **Name**, **Owner**, etc. in the appropriate fields.
4. Select **JDBC** as the **Application Type**. The required connection parameters are then displayed in the **Attributes** tab.

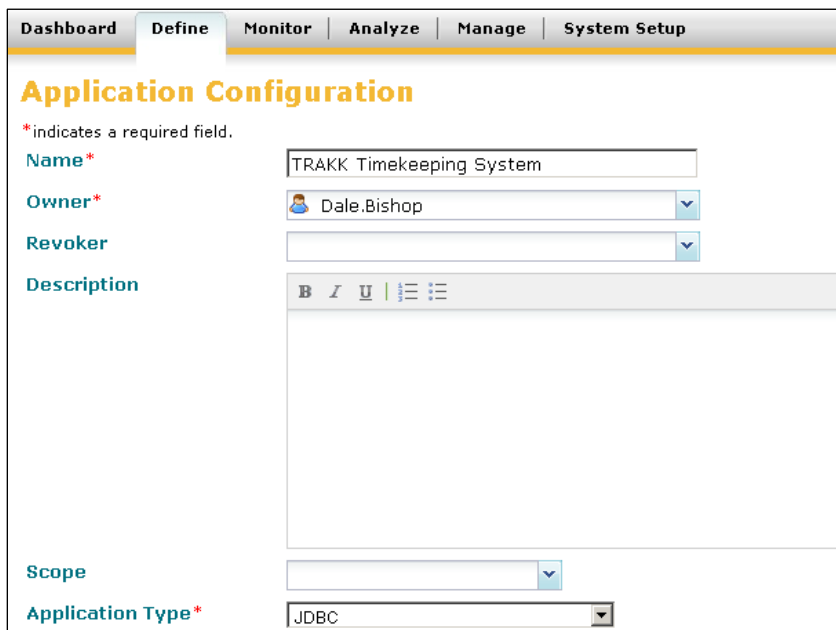


Figure 34: Application Type Specification

5. On the **Attributes** tab's Account page, specify the connection settings:
 - **Connection User**: user ID used in making the connection to the database; required
 - **Connection Password**: the password for the connection user account

- **Database URL:** URL for accessing the database; required
- **JDBC Driver:** appropriate JDBC driver to allow Java to interact with the database; required

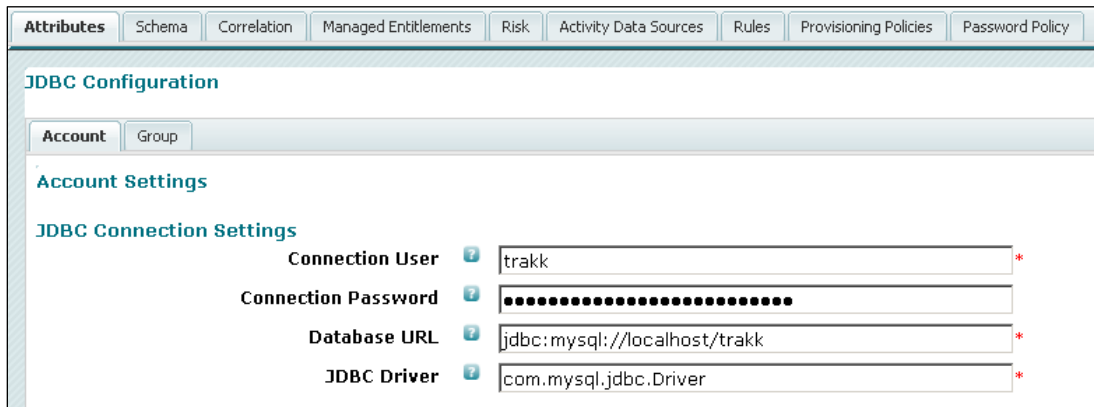


Figure 35: JDBC Connection Settings

- Specify the query settings to indicate what data should be retrieved from the database:
 - **SQL Statement:** SQL statement that selects the entire set of account records to aggregate into IdentityIQ
 - **useExecuteQuery:** flag indicating that `Statement.executeQuery()` should be invoked instead of the default `Statement.execute()` (rarely used but available when problems are encountered running specific SQL statements with `execute()` against a specific DBMS)
 - **getObject SQL:** SQL statement to retrieve *one* user account link (as opposed to the whole set retrieved by SQL Statement). This statement is executed for targeted re-aggregation following processing of an LCM provisioning request or a certification remediation to verify the requested change. The account-identifying value is specified with the special token `$(identity)` that is replaced with the requested `nativelIdentity` at runtime when the `getObject()` method is called on the JDBC connector. If this statement is not specified, the default query is **select [account schema attribute-name list] from [native object type] where [schema identity object] = [requested nativelIdentity]**. If that query will not accurately retrieve a record, the `getObject SQL` attribute must be specified.

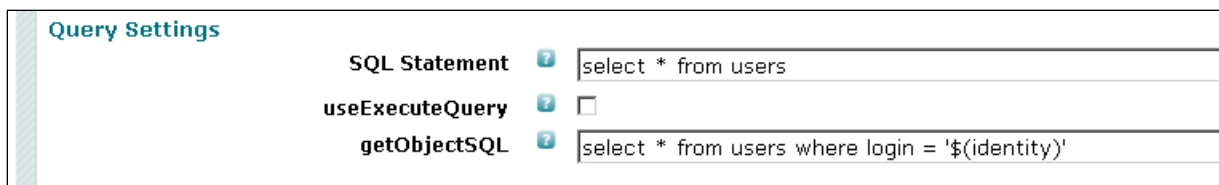
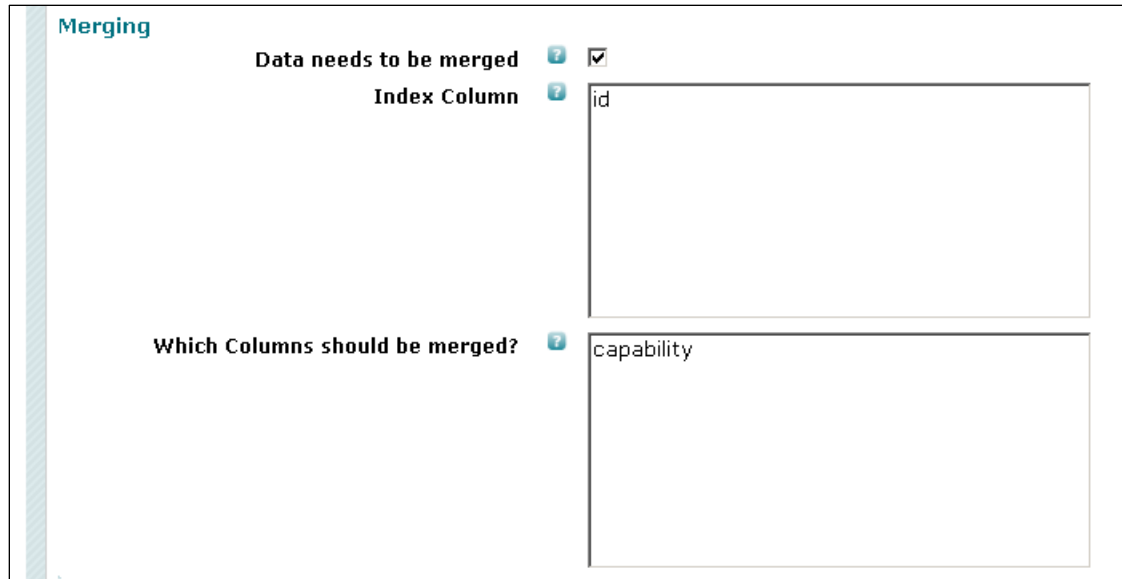


Figure 36: Query Specifications

- Specify the merging parameters to indicate whether data for a single object may span multiple lines that requires merging of data records:
 - **Data needs to be merged:** flag to enable merging of input records into a single object
 - **Index Column:** name of the column that is used to match records that need to be merged
 - **Which columns should be merged:** names of input columns from which values should be merged



Merging

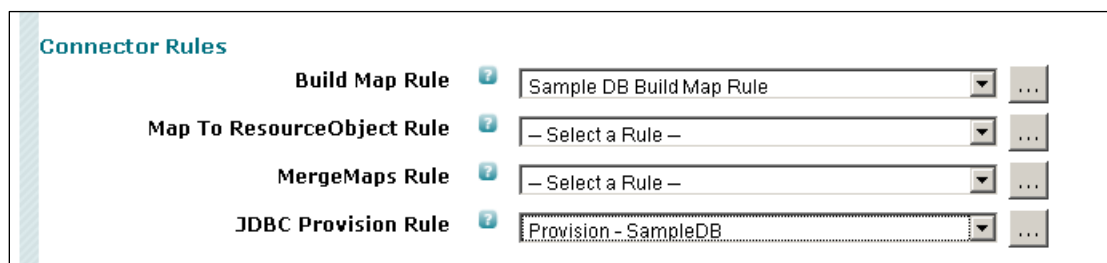
Data needs to be merged ☒

Index Column

Which Columns should be merged?

Figure 37: Merging Specification

8. Specify connector rules. These are beanshell code segments that are used to perform additional processing on data rows as they are read in or written out (depending on the rule). More details are provided on these rules in the *Connector Rules* section.
 - **Build Map Rule:** rule called for each data row read in to convert the string tokens into a java.util.Map object. If rule is not specified, the connector builds a map with the contents keyed by column name.
 - **Map to ResourceObject Rule:** rule called for each java.util.Map object created to convert the built object into a ResourceObject; if rule is not specified, the connector builds a ResourceObject using the schema
 - **MergeMaps Rule:** rule called when merging rows with matching index columns; rule receives existing map and newly parsed map that needs to be merged; if rule is not specified, the connector builds a combined java.util.Map using the original object and the merge attributes specified in the mergeColumns config option (the Which columns should be merged attribute).
 - **JDBC Provision Rule:** rule called to execute provisioning activity for the application; must be specified if provisioning is done for the application; returns a provisioningResult



Connector Rules

Build Map Rule

Map To ResourceObject Rule

MergeMaps Rule

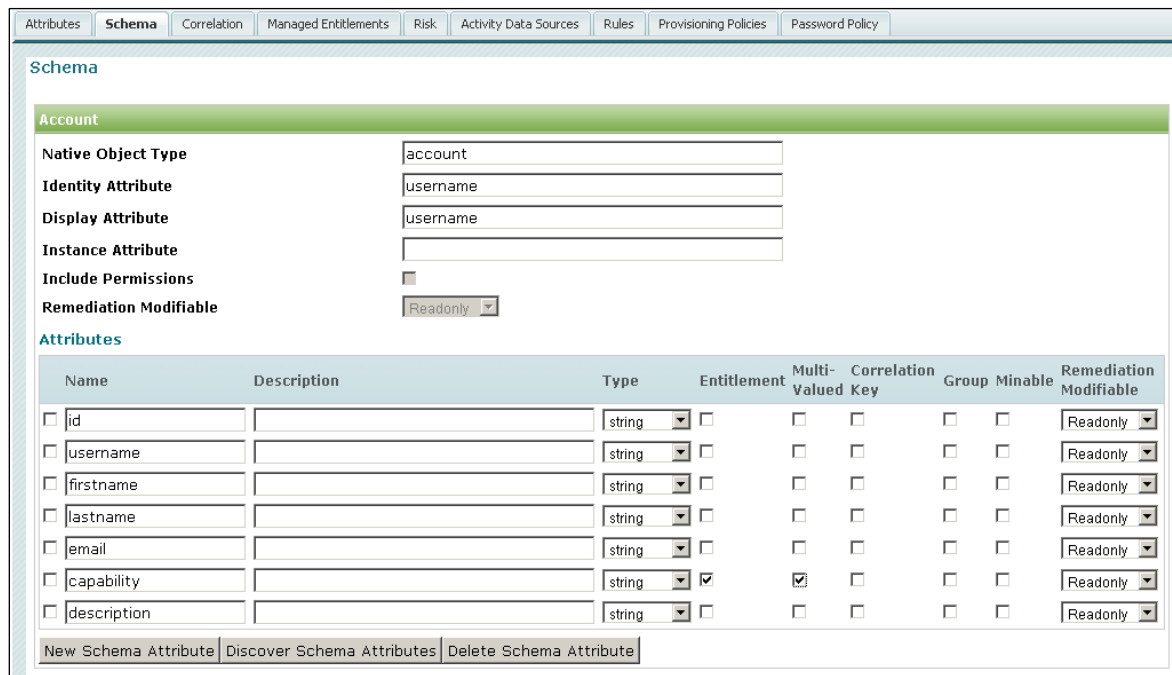
JDBC Provision Rule

Figure 38: Connector Rules Selection

9. On the **Groups** tab, specify the query settings, merging parameters, and rules specific to the application's account groups. These are specified in the same way as indicated in steps 5-8 above for

the account settings except their values are specific to groups instead of accounts. This is only applicable if group data is being aggregated from the JDBC application.

- Define the account (and group, if applicable) schema attributes on the **Schema** tab. Click **Add Account Schema** to manually create it or click **Discover Schemas** to retrieve it from the JDBC database automatically. If the Account Schema is manually created, clicking **Discover Schema Attributes** can automatically retrieve the attributes from the JDBC database. Mark any attributes as **Entitlement**, **Multi-Valued**, or **Group** attributes as appropriate.



Name	Description	Type	Entitlement	Multi-Valued	Correlation Key	Group	Movable	Remediation Modifiable
<input type="checkbox"/> id		string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> username		string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> firstname		string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> lastname		string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> email		string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> capability		string	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly
<input type="checkbox"/> description		string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Readonly

Figure 39: Account Schema

To define a group schema, click **Add Group Schema**, define the **Native Object Type**, **Identity Attribute**, and **Display Attribute** values, and click **Discover Schema Attributes**.

- Return to the Attributes tab and click **Test Connection** (at the bottom of the page) to verify whether IdentityIQ can connect to the database with the connection specifications provided. A message next to the **Test Connection** button indicates whether the connection was successful or not.

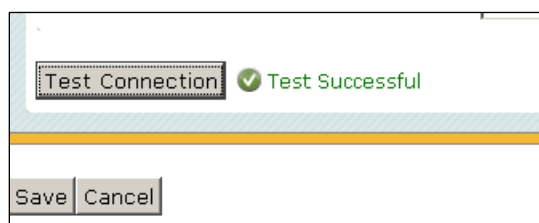


Figure 40: Successful Test Connection

NOTE: **Test Connection** tests both the connection and the SQL Statement. It will fail if the connection parameters are incorrect, if the SQL statement will not execute, or if the Account Schema has not been defined for the application.

12. Click **Save** to save the application definition.

Connector Rules

The connector rules allow for customization of how the connector reads data from and writes data to the JDBC database.

The first three rules are used during the aggregation process to manipulate the data records from the native columns to data fields that match the IdentityIQ schema for the application. The final rule (JDBC Provisioning Rule) specifies the provisioning logic for the connector.

The Map to ResourceObject Rule specifies how each attribute map should be transformed into a ResourceObject that will be used to create the application account (link) record.

Build Map Rule

The Build Map Rule allows additional map attributes to be populated based on the data read in each row. It is called for each data row read in (JDBC ResultSet) to convert the data into map of column name/value pairs. When no rule is specified, the connector executes its buildMapFromResultSet() method, which builds a map with the contents keyed by column name. Typically, the first statement in the rule executes that same method to extract the ResultSet data into a map from which the data elements can be easily extracted and examined. The rest of the rule evaluates the data elements and alters or creates additional map attributes based on the data values. For example, the native application might contains a “status” column with possible values “A” (active) and “I” (inactive); this could be used to map another boolean attribute “IIQDisabled” to False and True, respectively.

```
HashMap map = JDBCConnector.buildMapFromResultSet(result);
if ( schema.getObjectType().compareTo( Connector.TYPE_ACCOUNT ) == 0 ) {

    String active = map.get("status");
    if (active.equals("A")) {
        map.put( "IIQDisabled", false);
    } else {
        map.put("IIQDisabled", true);
    }
}
```

NOTE: IIQDisabled and IIQLocked are special attribute names that, when set, cause the account to be marked as disabled or locked in the UI.

MergeMaps Rule

In some cases, data for a single application account must be read in from multiple source records, generally recording one or more values into multi-valued attributes. Basic row merging involves matching records by their **Index Columns** (specified in the Merging section of the UI configuration) and combining the values in the merge columns into multi-valued attribute lists for each column named. Usually, the basic row merging is sufficient and no merge map rule is specified. A merge map rule is only needed when more complex logic is required to manipulate the data from its input format to the merged structure.

If a merge map rule is specified, it is called when the previous and current records read from the database should be merged. Both records are passed to the rule, along with the list of columns to be merged. The logic within the rule determines the merging procedure. The expected return value from the rule is the merged attribute map.

Map to ResourceObject Rule (Transformation Rule)

This rule is called after all the records have been read in and merged (if applicable). It is called for each Map object to convert it into a ResourceObject. A ResourceObject is a memory representation of the group or account. If this rule is not specified, the connector builds a ResourceObject using the schema. In most cases, the default behavior is the desired action and this rule is omitted. This rule is only necessary when additional data transformation is required to properly create the ResourceObject.

JDBC Provision Rule

Because JDBC databases have differences in SQL syntax and each specific database will have widely different table structures, there is no way for IdentityIQ to fully automate provisioning to them. Instead, it is up to the integration team to write the JDBC Provision Rule that manages the provisioning activities for the JDBC system. Like all IdentityIQ rules, this rule is written in beanshell. Its arguments are an Application, Schema, Connection (to the JDBC database), and ProvisioningPlan object and it returns a ProvisioningResult object. See the java docs for `sailpoint.object.Application`, `sailpoint.object.Schema`, `java.sql.connection`, and `sailpoint.object.ProvisioningPlan`, and `sailpoint.object.ProvisioningResult` for details on these object types.

Execution of the SQL statements or stored procedures to insert, update, or delete records on the JDBC database is managed through the appropriate JDBC method calls. The rule must examine the provisioningPlan passed to it and extract individual requests from it. Those requests then drive the execution of the appropriate JDBC calls.

ProvisioningPlan components may vary greatly from one to another. The table below illustrates many of the combinations of requests that can be included in a single provisioning plan. All requests in a single provisioning plan always relate to only one Identity.

Account Request	Attribute or Permission Request
Account Create	None
	Entitlement(s) Add
	Permission(s) Add
Account Modify	None (modify account attributes only)
	Entitlements Add, Modify, Remove combined in any permutation or one at a time
	Permissions Add, Modify, Remove combined in any permutation or one at a time
Account Deletion	None
Account Lock	None
Account Unlock	None
Account Enable	None
Account Disable	None

The rule should begin by retrieving the Account Requests. These are account level operations – create, modify, delete, disable, etc. The rule iterates through the list of account requests, checking the operation specified for it to determine the required action. Operations available for Account Requests are Create, Modify, Delete, Disable, Enable, Lock, and Unlock.

```
List accounts = plan.getAccountRequests();
for ( AccountRequest account : accounts ) {
    if (AccountRequest.Operation.Create.equals(account.getOperation())) {
        // Perform required action to create new account (e.g. insert record
        // in database)

    } else if (AccountRequest.Operation.Modify.equals(account.getOperation())) {
        // Perform required action to modify account (e.g. update SQL)

    } else if (AccountRequest.Operation.Delete.equals(account.getOperation())) {
        // Perform required action to delete account (e.g. delete record in
        // database)

    } else if (AccountRequest.Operation.Disable.equals(account.getOperation())) {
        // Perform required action to disable account (e.g. update SQL)

    } else if (AccountRequest.Operation.Enable.equals(account.getOperation())) {
        // Perform required action to enable account (e.g. update SQL)

    } else if (AccountRequest.Operation.Lock.equals(account.getOperation())) {
        // Perform required action to lock account (e.g. update SQL)

    } else if (AccountRequest.Operation.Unlock.equals(account.getOperation())) {
        // Perform required action to unlock account (e.g. update SQL)
    }
}
```

For certain operations (like Modify), the AccountRequest contains additional request information within an AttributeRequest. Attribute requests specify the account attribute to be impacted as well as the operation to be performed on it. Operations available for AttributeRequests are Set, Add, Remove, Revoke, and Retain.

```
} else if (AccountRequest.Operation.Modify.equals(account.getOperation())) {
    List mod_attr_requests = account.getAttributeRequests();

    if (mod_attr_requests != null) {
        for (AttributeRequest req : mod_attr_requests ) {
            if (req.getName().equals("capability")) {
                if (ProvisioningPlan.Operation.Remove.equals(req.getOperation())) {
                    // code to delete record inserted here

                } else if (ProvisioningPlan.Operation.Add.equals(req.getOperation())) {
                    // code to insert new record with the new capability inserted here
                }
            }
        }
    }
}
```

If the provisioning request is for permissions, rather than entitlement attributes, it will contain permissionRequests instead of AttributeRequests. Depending on the application and the request, it could potentially contain both. Provisioning requests would be navigated similarly to Attribute requests and has the same set of valid Operations. The permission request's right and target values (accessed through the getRight() and getTarget() methods) would be used in the SQL processed by the JDBC method call.

```
} else if (AccountRequest.Operation.Modify.equals(account.getOperation())) {
```

```
List mod_perm_requests = account.getPermissionRequests();

if (mod_perm_requests != null) {
    for (PermissionRequest req : mod_perm_requests ) {
        if (ProvisioningPlan.Operation.Remove.equals(req.getOperation())) {
            // code to delete record inserted here

        } else if (ProvisioningPlan.Operation.Add.equals(req.getOperation())) {
            // code to insert new record with the new capability inserted here
        }
    }
}
```

The rule must return a `ProvisioningResult`. Often, the rule encapsulates the processing in a try block, catching a `SQLException` to indicate failure and returning `Committed` when no exception is detected. The only applicable statuses for the JDBC connector are `Committed` and `Failed`.

```
try {
    ...
    [JDBC method call to execute SQL statement here]
    result.setStatus(ProvisioningResult.STATUS_COMMITTED);

} catch (SQLException e) {
    result.setStatus(ProvisioningResult.STATUS_FAILED);
    result.addError(e);
}
return result;
```

JDBC Governance Connector

Customers who have not yet upgraded to IdentityIQ version 5.2p1 or higher have the JDBC Governance connector available to them for collecting data from JDBC databases. Configuring this connector is identical to configuring the Direct connector, except the JDBC Provision Rule is not available and therefore does not need to be specified.

Document Revision History

Revision Date	Written/Edited By	Comments
April 2012	Jennifer Mitchell	Initial Creation (current version: 5.5)
Dec 2012	Jennifer Mitchell	Corrected copy error in JDBC connector Attributes tab info
Jan 2013	Jennifer Mitchell	Removed references to keystore attribute in LDAP Direct Connector section (SSL keystore info must be set at JVM level, not through connector attributes; this attribute existed in 5.5p2 but was not compatible with some app servers; this was fixed in 5.5p3)