|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Client Name | |  | | IdentityIQ  Environment Setup Instructions | |  | | Manjunath Madiraju  April 2020 Version: 1 | |  | |

**Table of Contents**

[Document Control 3](#_Toc38886613)

[1 Introduction 4](#_Toc38886614)

[2 Installing IdentityIQ onto a Server 5](#_Toc38886615)

[2.1 Prerequisites and Required Supporting Software 5](#_Toc38886616)

[2.2 Creating a Directory to Hold Installer Programs 5](#_Toc38886617)

[2.2.1 Required/Strongly Recommended Applications 6](#_Toc38886618)

[2.2.2 Optional Recommended Applications 6](#_Toc38886619)

[2.3 Installing the Java JDK/JRE 6](#_Toc38886620)

[2.4 Installing Apache Tomcat 9](#_Toc38886621)

[2.5 Installing the Notepad++ Windows Application 14](#_Toc38886622)

[2.6 Installing the IdentityIQ Application 15](#_Toc38886623)

[2.7 Configure Extended Searchable Attributes 17](#_Toc38886624)

[2.8 Configuring Database 18](#_Toc38886625)

[2.9 Configure the IdentityIQ Installation 19](#_Toc38886626)

[2.10 Initializing and Launching IdentityIQ 20](#_Toc38886627)

[2.10.1 Import the initial configuration objects into the IdentityIQ database 20](#_Toc38886628)

[2.10.2 Launch IdentityIQ 20](#_Toc38886629)

[2.11 Enabling Lifecycle Manager (LCM) 21](#_Toc38886630)

[2.12 Configuring IdentityIQ 22](#_Toc38886631)

[2.13 Load Balancer 29](#_Toc38886632)

[2.14 Setup SSL 30](#_Toc38886633)

[2.15 System Administrator Capability 30](#_Toc38886634)

[The spadmin account is the most powerful account of the system, like root on a Unix system or Administrator on Windows. A common recommendation for securing a Windows installation is to create an alternate administrative account and disable the original. This recommendation is useful for IdentityIQ as well. In case an alternate spadmin account is created, make sure to also make this account protected by setting the protected attribute on the object. This will ensure that the account is never accidentally deleted:<Identity name="altadmin" password="\*\*\*\*" protected="true"> 30](#_Toc38886635)

[2.16 Encryption 31](#_Toc38886636)

[2.17 Re-Encrypt Passwords 32](#_Toc38886637)

[2.18 Object Level Auditing 35](#_Toc38886638)

[2.19 Domain Account for secure operating system 35](#_Toc38886639)

[2.20 Configuring IdentityIQ to communicate with Active Directory over SSL 36](#_Toc38886640)

[2.21 Generate IQService Server Certificate 36](#_Toc38886641)

[2.22 Configuring IdentityIQ to communicate with Active Directory over SSL 38](#_Toc38886642)

[2.23 Install and Register the IQService for Use with Windows 39](#_Toc38886643)

[2.24 Install Lync Administrative Tools if required 40](#_Toc38886644)

[2.25 Microsoft SQL Server Client 44](#_Toc38886645)

[2.26 SSD Setup 44](#_Toc38886646)

[2.27 Setup Deployment Accelerator 56](#_Toc38886647)

[2.28 Install Git and GitLab Runner 57](#_Toc38886648)

[2.28.1 Command line instructions to Download Repository 59](#_Toc38886649)

# Document Control

This is a version-controlled document. The control and release of this document is the sole responsibility of the document owner.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version control** | | | |
| Version | 1.0 | Date | 04/24/2020 |
| Classification | Confidential | Author | Manjunath Madiraju |
| Document Title | IdentityIQ Environment Setup Instructions | | |
| Approved by |  | | |
| Released by |  | | |
|  | | | |
| **Owner details** | | | |
| Name | Manjunath Madiraju | | |
| Office |  | | |
| Contact number | 925.699.9716 | | |
| Email Address | [mmadiraju@kpmg.com](mailto:mmadiraju@kpmg.com) | | |
|  | | | |
| **Revision history** | | | |
| **Issue** | **Date** | **Author** | |
| Draft 1.0 | 04/24/2020 | Manjunath Madiraju | |
|  |  |  | |

# Introduction

This document covers the steps required to install IdentityIQ onto a host computer, as a sandbox environment. This is not intended as a guide for installing a production environment. The intended audience is system administrators or SailPoint team members who may need to install IdentityIQ on a server or in a laboratory environment like a virtual machine.

This document shows examples where the utilities and tools are installed to the C: drive of a Windows-based system using the Apache Tomcat application server package. The examples shown here are for illustrative purposes only; it is recommended that the SailPoint directory and the installer files directory be placed on a hard drive other than the C: drive when deployed for production use.  Installing to a non-C: drive letter prevents log files from potentially filling up the host operating system's drive.  In your environment please be consistent by installing all the utilities to the E:, F:, H:, or whatever drive is available for application software in your virtual machine architecture.

This document includes screen shots of various installation steps, including dialog boxes, Windows Explorer directories, etc. Note that Windows 7, Windows 10, and Windows Server 2012 will all look slightly different, and the graphics and click-through trails you see may not exactly match the ones illustrated here. However, the steps are analogous on most versions of Windows.

# Installing IdentityIQ onto a Server

## Prerequisites and Required Supporting Software

This section covers the procedures to be followed when installing IdentityIQ onto a new server, including the installation of supporting software packages for the deployment environment.

In this document, we will install IdentityIQ in a Windows Server 2008 R2 (64 bit) operating system, using Apache Tomcat as the application server, and a MySQL database.  This combination is common for a sandbox environment, however Linux based sandbox combinations are also very common.  Details on other supported platforms are in the SailPoint IdentityIQ Installation Guide.

The following prerequisites must be in place before attempting to install IdentityIQ onto a server or virtual machine:

* The server’s Fully Qualified Domain Name (FQDN) must be known and documented.
* The server must be online with its operating system already installed.
* Your desktop account must have remote desktop protocol (RDP) access to the server.
* Your account must have administrative privileges to the server to which IdentityIQ is being installed.
* The target server must have at least 15GB of free space on the C:\ drive  for software installation.
* The target server should have at least 2GB of RAM available, 8GB of RAM is preferred.

## Creating a Directory to Hold Installer Programs

The first order of business is to create a "SailPoint" directory to hold all of the IdentityIQ related packages and to create a place to hold the installer programs for the supporting applications that must be installed with IdentityIQ.  Creating a directory to house the installer saves time in the future by providing a quick and easy location to install the supporting applications on other sandbox hosts. If you are configuring a sandbox host to use as part of a deployment project, consider adding the "InstallerFiles" directory to your source control or change control system's directory structure.  In this example we will simply keep all the resources in one place.

* Create a directory off of the F:\ drive called "SailPoint" and a sub-directory inside of that one named “InstallerFiles” to hold the installation images.



After the directory is created to house the installer programs, you can download or copy the necessary installers. The list below shows installer file names and tool names for various packages needed for an IdentityIQ Dev. These versions are current at the time of this writing, but there may be more current versions available when reading this guide. Also note that this document only describes how to install the essential applications, and does not provide installation details for every one of these recommended applications. Check the prerequisites and supported versions for the version of IdentityIQ you are installing before proceeding.

### Required/Strongly Recommended Applications

These applications are required, or strongly recommended, for a baseline install.

* Base **IdentityIQ product installer**  (identityiq-7.2.zip)
* Latest **IdentityIQ Patch** (identityiq-7.2p2)
* **Java 64bit JDK**  (jdk-8u181-windows-x64)
* **Apache Tomcat Servlet Container** (apache-tomcat-8.5.34.exe)
* **Notepad++** - A free, easy-to-use Text Editor, downloadable at [https://notepad-plus-plus.org/](https://community.sailpoint.com/notepad-plus-plus.org)
* **7-zip for Windows**: an open source zip utility for Windows (this is optional, and is only needed if no native ZIP/gz utility is available). It can be downloaded at [http://www.7-zip.org/](https://community.sailpoint.com/www.7-zip.org)

### Optional Recommended Applications

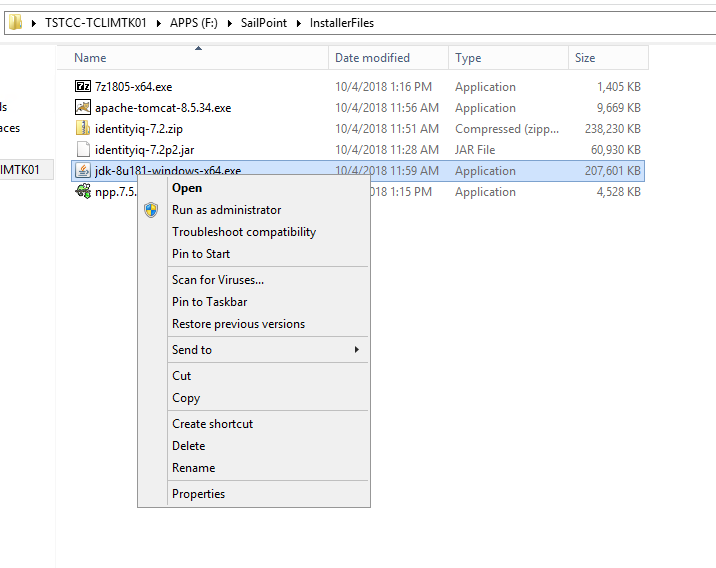
These applications may be useful as you use IdentityIQ; however, they are not required for a baseline install, and are not discussed in detail in this document.

* **Eclipse Java Development IDE** ([https://www.eclipse.org/downloads/](https://community.sailpoint.com/www.eclipse.org/downloads)). Running the Eclipse IDE lets you make use of the [IdentityIQ Deployment Accelerator](https://community.sailpoint.com/docs/DOC-3384).
* **Tortoise-GIT Git client**: an open-source Windows shell interface to Git, based on TortoiseSVN, downloadable at [https://tortoisegit.org/](https://community.sailpoint.com/tortoisegit.org)
* **Apache Directory Studio**: a directory tooling platform that can be used with any LDAP server, particularly with ApacheDS, downloadable at [https://directory.apache.org/studio/](https://community.sailpoint.com/directory.apache.org/studio)
* **PuTTy**:  A free, open-source terminal emulator, serial console and network file transfer application, downloadable at www.putty.org.
* **WinSCP**: A free open source SFTP client, FTP client, WebDAV client and SCP client for Windows. It also offers scripting and basic file manager functionality. It is downloadable at [https://winscp.net/eng/download.php](https://community.sailpoint.com/winscp.net/eng/download.php)

## Installing the Java JDK/JRE

IdentityIQ requires a Java environment be present on the server it is installed to.  To support the deployment utilities for automating deployment and customizations to IdentityIQ, we use a JDK (developer kit) edition of Java.  For Windows systems the recommended JDK is the Oracle/Sun version 1.8 JDK of the most recent patch level.

After downloading the JDK and installer and saving it to the "InstallerFiles" directory, proceed with installing the JDK by right clicking on the JDK installer file and selecting “Run as administrator”.



If prompted for a “User Access Control”, click “Yes” to allow the installer to continue executing.  When prompted, click “Next” to begin the installation wizard.

The installation wizard will prompt you for what components to install and what directory location to install the JDK package into.  Leave all the factory defaults, including the target installation directory which should resemble “C:\Program Files\Java\jdk1.8.0\_45” .  Click “Next” to advance the installation wizard.

The wizard will proceed with the installation of the Java Development Kit software.  When prompted for where to install the Java Runtime Environment, leave the default path and click “Next” to continue the installation.

The installer will proceed with the installation of the JRE software.  When prompted, click “Finish” to exit the installer wizard.

Close any browser sessions that the installer for Java opens after completing the installation.  After the installation of Java is complete you need to update the Windows server’s environment variables to include a JAVA\_HOME environment variable that points to the JDK installation and to include the JDK “bin” directory in the system PATH.

To do this, browse on the start menu to the “Computer” icon and right click and select “Properties”.

Click on the “Advanced System Settings” link in the Systems details window.

Next, click on the “Environment Variables” button at the bottom right of the “System Properties” window.

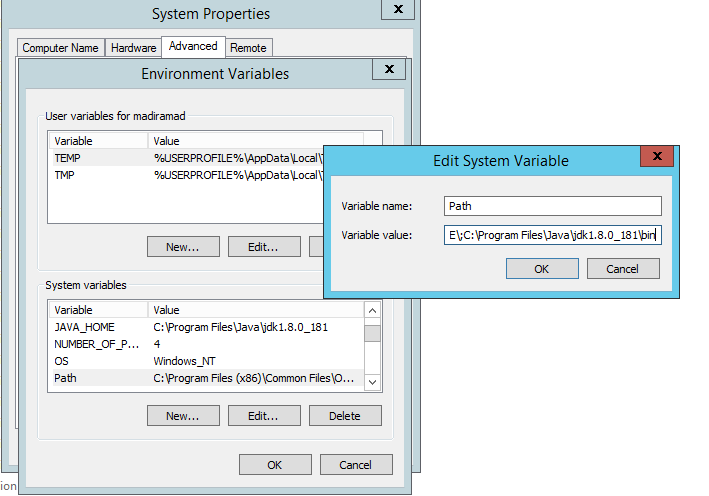
In the “System variables” section of the “Environment Variables” window, click on the “New…” button to define a new system-wide environment variable.

In the “New System Variable” window that appears enter “JAVA\_HOME” (without the quotes) for the variable name and “C:\Program Files\Java\jdk1.8.0\_45” (again, without the quotes) for the variable value.  Note: Your exact path may be different; be sure to copy the path from Windows Explorer into this dialog box to avoid typos.



Click “OK” to save the new JAVA\_HOME system environment variable.  Next, scroll through the list of system variables and find the “Path” system environment variable.  Highlight it and click “Edit”.

This will bring up an “Edit System Variable” window for the Path variable.  Place the cursor in the “Variable value” text box and press the “End” key to advance to the end of the text.  With the cursor at the end of the string of text, append “;C:\Program Files\Java\jdk1.8.0\_45\bin” (with the leading semicolon, but without the surrounding quotes) to the text in the box.  Again, your specific path may be different; copy and paste to prevent errors and typos.



Click “OK” to save the updated Path system environment variable.  Click “OK” to dismiss the “Environment Variables” window.  Click “OK” to dismiss the “System Properties” window.  Finally, click the close-window “X” in the top-right to dismiss the System details window.

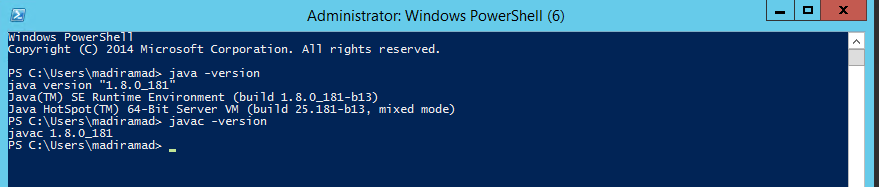
Now you can test both environment variables to make sure they were configured correctly, and test the installation of the JDK and JRE files to make sure they run properly.  From the start menu click on “Command Prompt” to open a command window.

In the command window type two commands:

javac -version

java -version

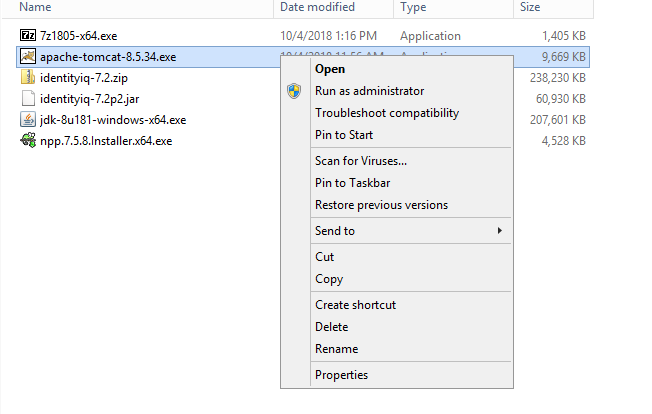
These should parrot back the installed version of the Java tool kit, as shown in the example screen capture



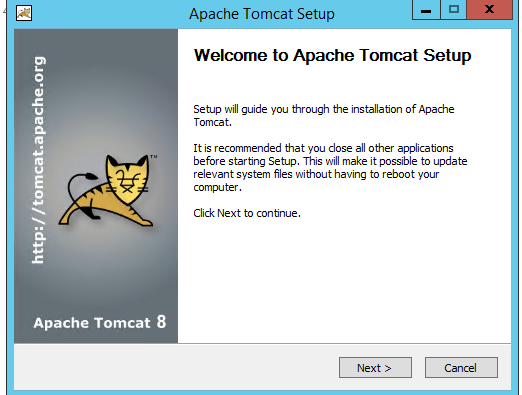
When “java” and “javac” both run successfully you have successfully installed Java on the server.

## Installing Apache Tomcat

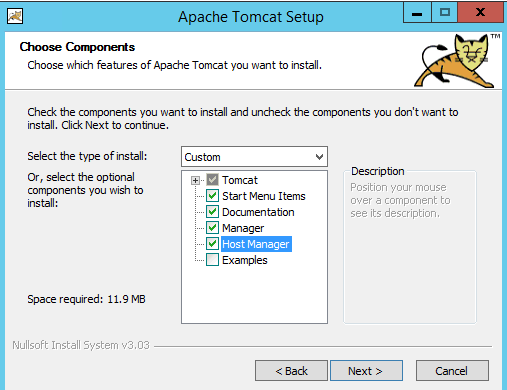
After the Java JRE and JDK packages have been installed, you can install Apache Tomcat.  Java is a prerequisite for Tomcat and must be installed first.  To install Tomcat, begin by right-clicking on its installer executable in the “InstallerFiles” directory and select the “Run as administrator” option.



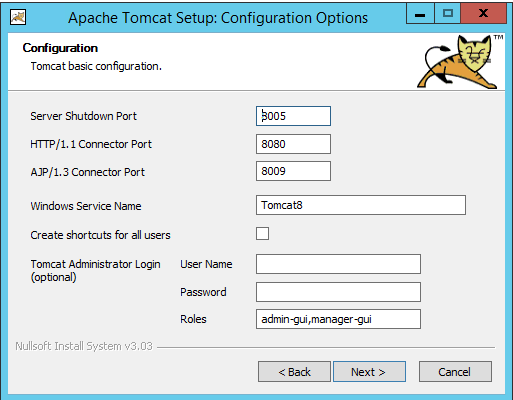
If prompted with a User Account Control dialog select the “Yes” option to allow the installer to continue executing.  On the welcome screen for the Apache Tomcat Setup Wizard click “Next” to advance to the start of the installation wizard. When presented with the “License Agreement” click the “I Agree” button to advance to the next step of the installation wizard.



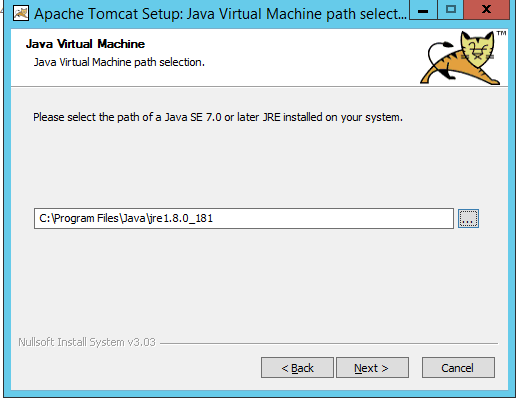
On the “Choose Components” screen of the installation wizard select the “Host Manager” option check box in addition to the factory defaults and click “Next” to advance the installation wizard.

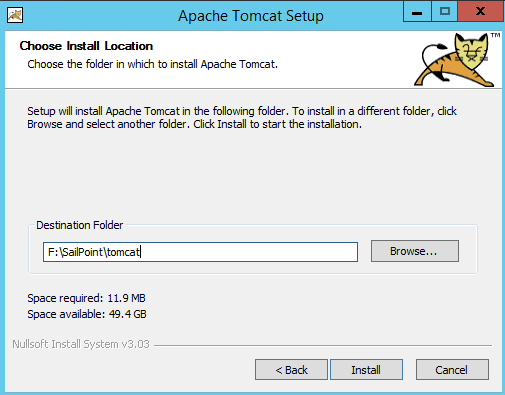


On the “Configuration” screen leave the factory defaults for all fields (server shutdown port, HTTP 1.1 port, etc.) and click “Next” to continue.



When prompted for the path to the Java Virtual Machine, verify that the default path “ C:\Program Files\Java\jre1.8.0\_181” matches the directory where the JRE was installed during the installation of Java.  This step specifies which JRE version on the server is used to run Tomcat (multiple Java versions can be installed simultaneously). Click “Next” to advance the installation wizard.   On the “Choose Install Location” step change the default Destination Folder target to  “\tomcat” and click “Install” to advance the installation wizard.

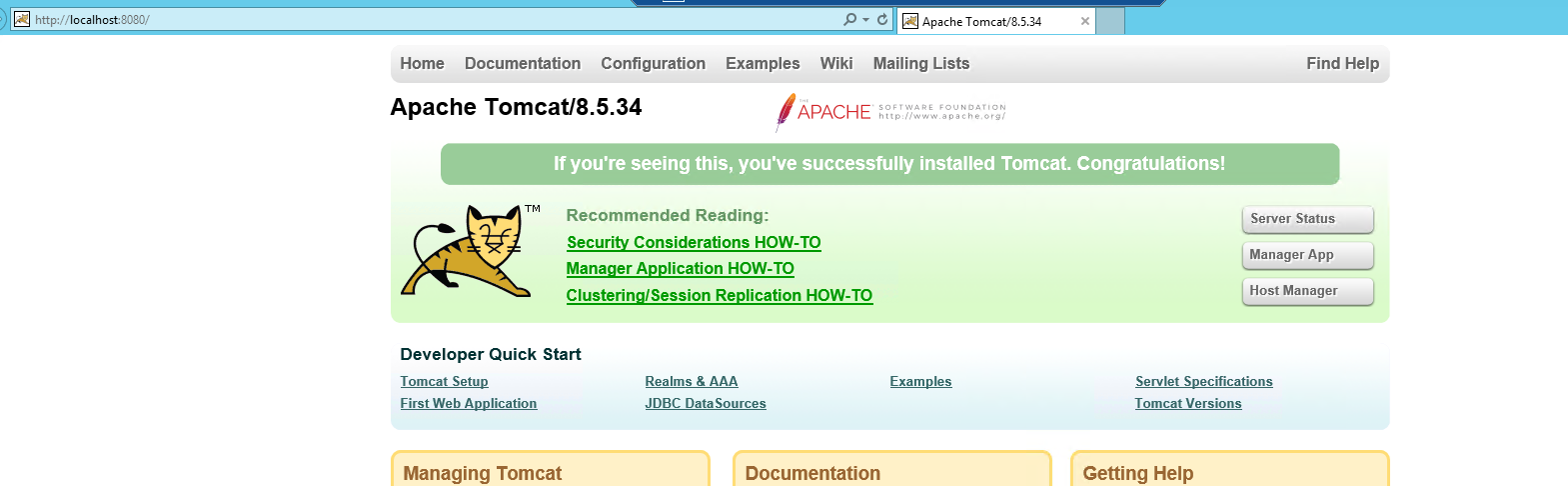




The installation wizard will proceed with installing Tomcat to the destination folder.   When prompted with the “Completing the Apache Tomcat Setup Wizard” screen, un-check the “Show Readme” check box, leave the “Run Apache Tomcat” checkbox selected and click “Finish” to close the Tomcat installation wizard.

You may briefly see the “Apache Commons Daemon Service Manager” starting the Tomcat service on the host after clicking “Finish”.

At this point, the installation of Apache Tomcat has run on the host. The next step is to test Tomcat to make sure it is functioning correctly.  Open Internet Explorer from the start menu of the server.  In the address bar of Internet Explorer type the following URL: http://localhost:8080 and press enter.  You should see a confirmation page from Apache Tomcat load in the browser.  This page confirms Tomcat is installed correctly.

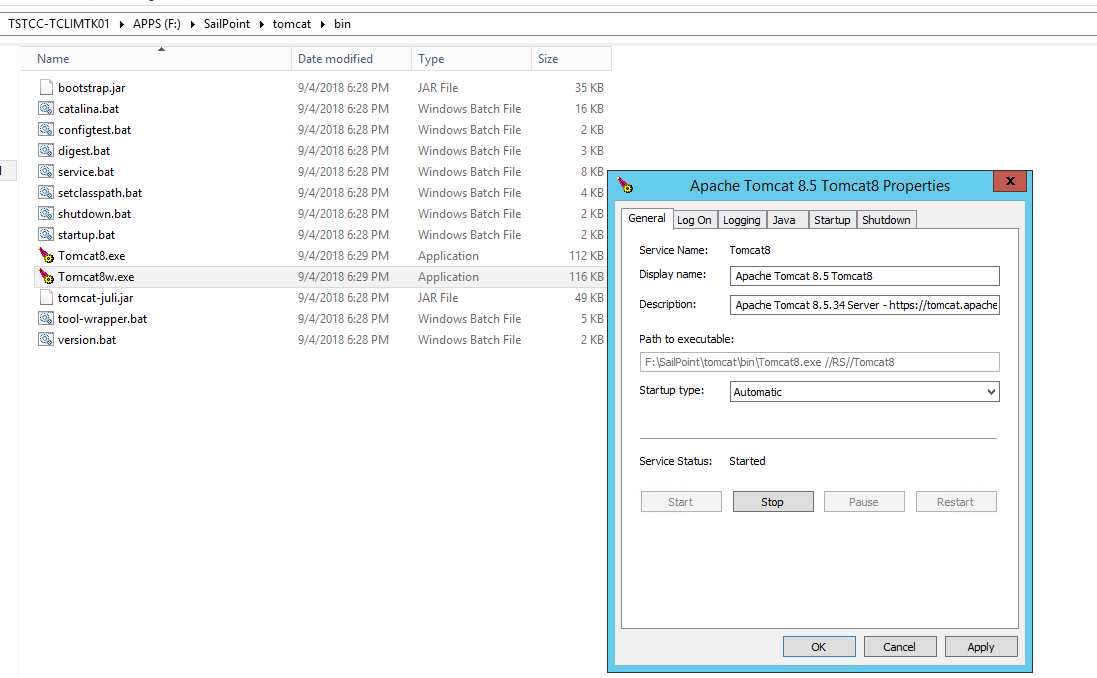


Next, you will apply some settings to Tomcat to make its configuration optimal for IdentityIQ.  Open Windows Explorer (note: not Internet Explorer) and browse to the following directory:

\tomcat\bin

In this directory you will see a program called “Tomcat8w.exe”.  This is the Tomcat 8 administration tool for Windows.  Right click on this program and select “Run as administrator”.

If prompted for a User Account Control, click the “Yes” option to allow the administration tool to launch.   The administration tool will load a window like this:



Select the “Startup Type” of “Automatic” and click “Apply”.  This configures Tomcat to automatically start when Windows restarts.  This way the server can be rebooted and IdentityIQ can automatically come back on line.

Next, click on the “Java” tab of the administration tool.  At the bottom of the “Java Options” box add the following lines.  These enable compatibility with Sun Java, extend some memory options, and allow compatibility with the Sun SOAP parser for interfacing with Remedy ticketing systems.

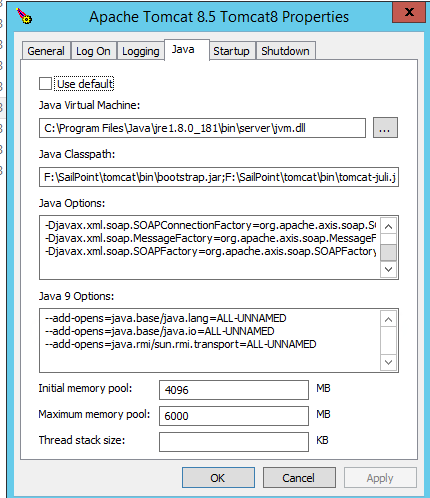
-Dsun.lang.ClassLoader.allowArraySyntax=true

-Djavax.xml.soap.SOAPConnectionFactory=org.apache.axis.soap.SOAPConnectionFactoryImpl

-Djavax.xml.soap.MessageFactory=org.apache.axis.soap.MessageFactoryImpl

-Djavax.xml.soap.SOAPFactory=org.apache.axis.soap.SOAPFactoryImpl

In the “Initial memory pool” text box enter “4096” without the quotes.   In the “Maximum memory pool” text box enter “6000” without the quotes.  These parameters set the amount of RAM Java utilizes at startup and maximum load.  After keying in (or copy/paste) these items click the “Apply” button.  If your sandbox environment has less than 4GB of RAM available than an initial memory pool of "1024" and a maximum memory pool of"2048" can be substituted.  These smaller values will work, but will not support larger-scale data sets under IdentityIQ.  Your sandbox server should have at least 2GB of RAM available to the host operating system to run MS SQL, Tomcat, and the IdentityIQ application.



Finally, click back to the “General” tab.  Click the “Stop” option to stop the Tomcat 8.5 service.  After the service has stopped (it will take a few seconds) click the “Start” option to re-start the service.   Click “OK” to dismiss the Tomcat 8.5 administration tool.  Open Internet Explorer and again browse to [http://localhost:8080](https://community.sailpoint.com/localhost:8080) to verify that Tomcat restarted successfully.

## Installing the Notepad++ Windows Application

Many tasks for administrating and debugging IdentityIQ revolve around reading or editing text files.  The default text editors for Windows (WordPad and Notepad) often fall short on features required by administrators.  Notepad++ is a free and recommended tool that can be used to change IdentityIQ’s configuration files, read its log files and modify the XML formatted import/export files that IdentityIQ uses.  Its use alongside Windows-based IdentityIQ deployments is so common that installing it is very strongly recommended, and is thus covered here so that it can be consistently installed on every server running IdentityIQ.

Locate the Notepad++ installation file in the “InstallerFiles” directory on the target server.  Right click on the installer and select the “Run as administrator” option.

If prompted for a User Account Control, select the “Yes” option to allow the Notepad++ installer to continue.  Select “English” as the language for Notepad++ and click “OK” to continue.  Click “Next” on the first page of the Notepad++ installer to start the installation wizard.

Leave the default target directory, “C:\Program Files (x86)\Notepad++” in place and click “Next” to continue the installation.

On the first “Choose Components” screen leave all of the defaults in place and click “Next” to continue with the wizard.  On the second “Choose Components” screen leave all the defaults un-checked and click “Install” to proceed with the installation.

The installation wizard will complete the installation of the package.  When prompted un-check the “Run Notepad++” checkbox and click “Finish” to complete and exit the installation program.

## Installing the IdentityIQ Application

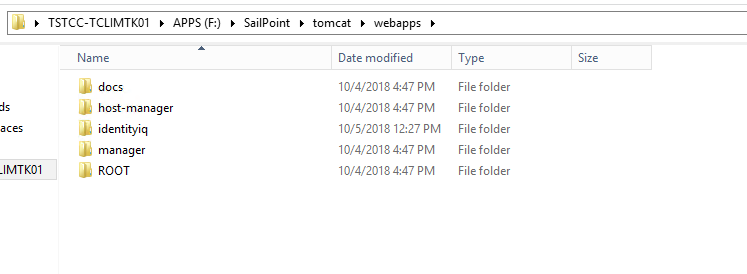
IdentityIQ is installed as a “web application” under the Apache Tomcat “webapps” directory.  The application binaries are installed to a specific directory under Tomcat and from that directory Tomcat serves up pages to users who request activity through their browsers.  Tomcat also loads the IdentityIQ scheduler and background processes for scheduled tasks that support IdentityIQ.

The installation process for IdentityIQ under Tomcat involves creating a web application directory.  After that, the core product binaries are installed into the directory.  Then any patch binaries are overlaid on top of the core product binaries.  Next the configuration file, “iiq.properties” is modified to direct IdentityIQ to the database that is used to store its backing data.  Finally the application server is restarted after all of these changes are applied to bring the instance of IdentityIQ online.  An additional step, covered in the Enabling Lifecycle Manager (LCM) section of this document, is used to load configurations and customizations onto an installed copy of IdentityIQ.

First you must create the web application directory on your Tomcat server.  Browse to the following directory using Windows explorer:

\tomcat\webapps

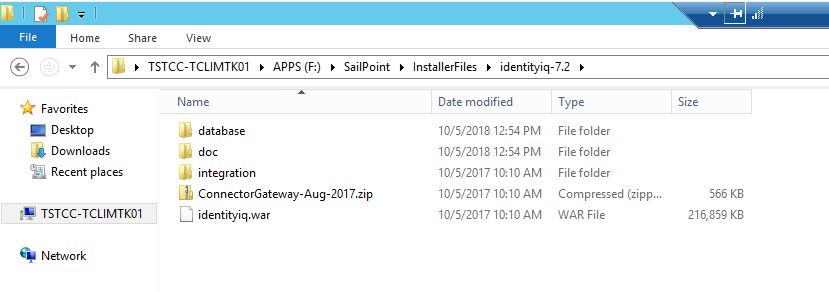
This is the directory that houses the web applications loaded under a Tomcat installation. Create a new directory called “identityiq” (all lower case) as illustrated below.



This directory provides a target location to install the IdentityIQ application binaries into.  Next locate the “ identityiq-7.2.zip” file that should be located in the “InstallerFiles” directory of the target server.  This file is the base product installer image.  Right click on this file in Windows Explorer and select the “Extract All…” option.

This will open the Windows Zip file extraction tool.  Leave the default extraction path of “ \InstallerFiles\identityiq-7.2 ” as the destination folder and click the “Extract” button.

The zip file will require a few seconds to completely unzip itself.  Windows will automatically open an explorer window into the newly created “identityiq-7.2” directory.  This directory contains documentation, integration notes, and most importantly the “identityiq.war” (web application archive) file that contains the IdentityIQ application binaries.



Note the path to the .war file shown above.  It should be “F:\InstallerFiles\identityiq-7.2\identityiq.war”.  Next, open up a command prompt window by right clicking on “Command Prompt” from the start menu and selecting the “Run as administrator” option.  If prompted for a User Account Control, select the “Yes” option to allow the command prompt to open.

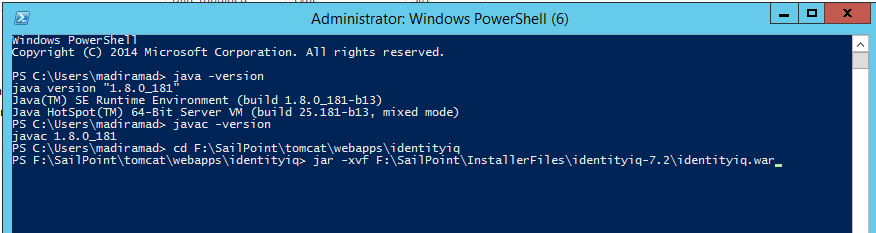
Next, in the command prompt window, change directory to the web application directory for IdentityIQ by running the command:

cd \tomcat\webapps\identityiq

Next, un-jar (the Java equivalent of un-Zip) the IdentityIQ war file into the web application directory by running the following command:

jar -xvf \InstallerFiles\identityiq-7.2\identityiq.war

The following screen capture shows the command immediately before execution.



The war file will spend several seconds extracting itself into the “…\webapps\identityiq” directory.

This lays out the 7.2 GA (generally available) application binaries.

The next step is to install the latest patch binaries on top of the base GA release.  To do this we un-jar the patch jar file into the web application directory so that the files from the patch overwrite the files in the GA release. (In the **Creating a Directory to Hold Installer Programs** section at the start of this document, the patch jar file (identityiq-7.2p2.jar) was listed as one of the files to copy to the “InstallerFiles” directory of the target server.)

If you're installing a patch after you've completely installed IdentityIQ and begun using it (that is, after you've set up your database and initialized IdentityIQ, as described in later steps in this document), refer to the readme for the patch for full installation/upgrade instructions. You may need to upgrade your database and run an upgrade script to complete the patch installation.

Keeping the same command prompt open, run the following command to overlay the patch files on top of the GA files. (Note: if installing a patch newer than 7.2p2 substitute the name of your higher patch file for the one shown here.)

jar -xvf  \InstallerFiles\identityiq-7.2p2.jar

The patch file will spend a few moments un-zipping itself in the web applications directory. Once the patch file is unzipped, the patch installation is complete.

## Configure Extended Searchable Attributes

1. Using the File Browser, locate the file: **tomcat\webapps\identityiq\WEB-INF\classes\sailpoint\object\IdentityExtended.hbm.xml**

Edit the file using any editor:

2. We know that we will need additional searchable and indexed extended identity attributes, but we don’t yet know which ones. For these, we will specify 6 placeholder attributes.

<property name="extended1" type="string" length="450"

index="spt\_identity\_extended1\_ci"/>

<property name="extended2" type="string" length="450"

index="spt\_identity\_extended2\_ci"/>

<property name="extended3" type="string" length="450"

index="spt\_identity\_extended3\_ci"/>

<property name="extended4" type="string" length="450"

index="spt\_identity\_extended4\_ci"/>

<property name="extended5" type="string" length="450"

index="spt\_identity\_extended5\_ci"/>

<property name="extended6" type="string" length="450"/>

<property name="extended7" type="string" length="450"/>

<property name="extended8" type="string" length="450"/>

<property name="extended9" type="string" length="450"/>

<property name="extended10" type="string" length="450"/>

<property name="client-identity-1" type="string" length="450"

access="sailpoint.persistence.ExtendedPropertyAccessor"

index="spt\_identity\_clientidentity1\_ci"/>

<property name="client-identity-2" type="string" length="450"

access="sailpoint.persistence.ExtendedPropertyAccessor"

index="spt\_identity\_clientidentity2\_ci"/>

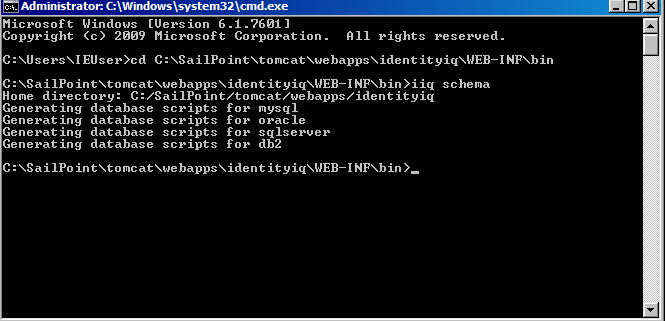
etc.,

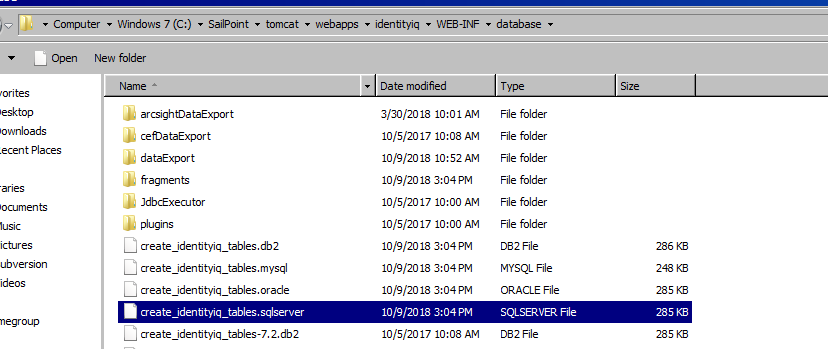
1. Save the changes to the file.

## Configuring Database

Next, generate the database schema for IdentityIQ:

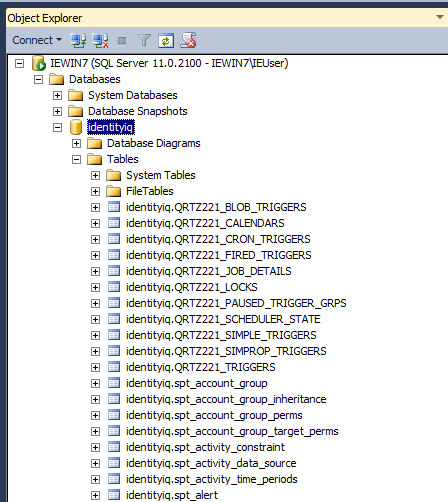
1. Using a command window, navigate to the \tomcat\webapps\identityiq\WEB-INF\bin directory.
2. Run the following command from the command line to generate the database schema files:  
     
   iiq schema





1. execute the IdentityIQ DDL scripts from create\_identityiq\_tables

Validate if the **identityiq** database is created and if tables are created successfully.



## Configure the IdentityIQ Installation

* 1. Configure IdentityIQ to use the database you created.Access the iiq.properties file and update the following information for the application and plugin databases:—

Host Name—

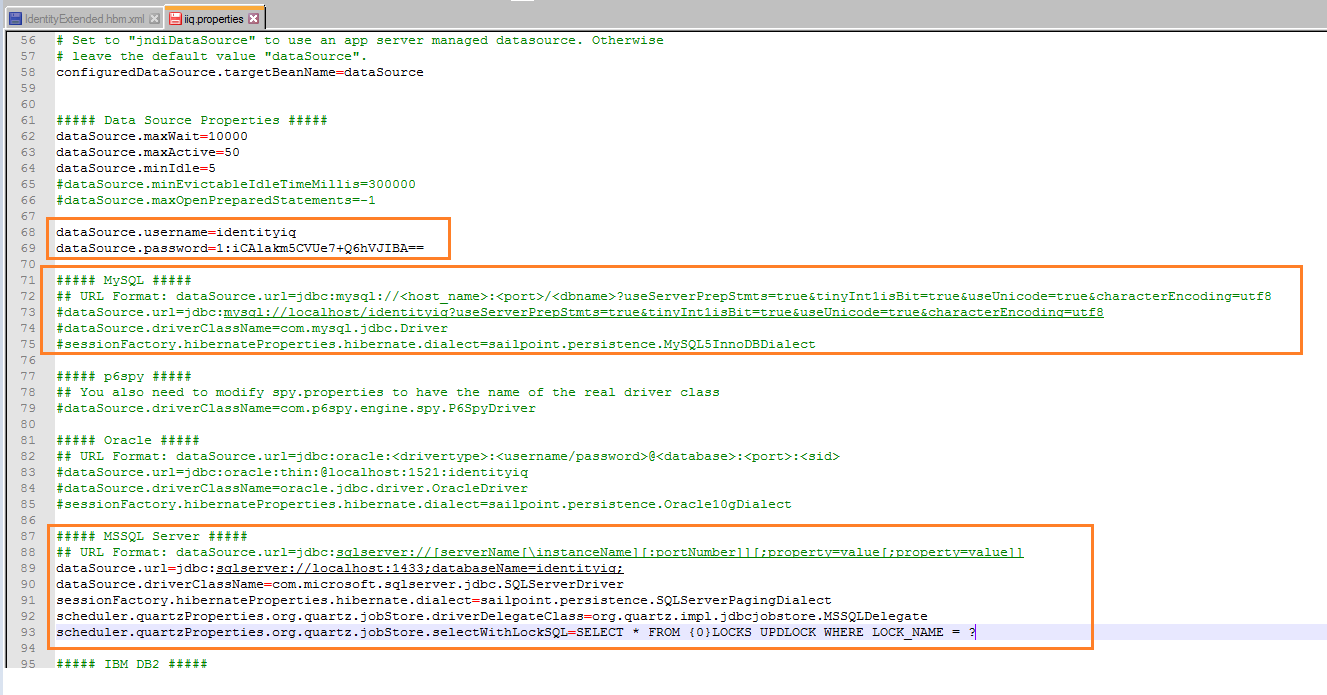
Database Type—

Database Name—

User ID—

Password

The iiq.properties file is located in **\tomcat\webapps\identityiq\WEB-INF\classes**\.The password for the database connection can be clear text or encrypted. It is highly recommended that you use an encrypted password.To create an encrypted password, run the iiq encrypt <password> command from the identityiq\_home\WEB-INF\bin directory.



* 1. Import the initial configuration objects into the IdentityIQ database.Launch the IdentityIQ console by running the iiq console command from the identityiq\_home\WEB-INF\bin directory. Use the console command import init.xml to import the configuration objects as showniiq console> import init.xml> quit
  2. Using a database client, execute the upgrade\_identityiq\_tables-7.2p2 DDL script in WEB-INF/database that is appropriate for your database type. The database scripts are cumulative, so you will receive errors for the schema changes that have already been applied in previous patches.
  3. Apply the patch by using a command line interface to execute the command

"iiq patch 7.2p2" (without the quotes) from the **\tomcat\webapps\identityiq\**WEB-INF\bin directory.

* 1. Start your application server.

## Initializing and Launching IdentityIQ

### Import the initial configuration objects into the IdentityIQ database

1. Open a command prompt.
2. Navigate to \tomcat\webapps\identityiq\WEB-INF\bin
3. Launch the iiqconsole by running this command.   
     
   iiq console  
     
   Allow the console a few moments to launch. The console is running when you see a > symbol in the command prompt.
4. In the IdentityIQ console, run this command:  
     
   import init.xml

1. When the import is complete, type quit to exit from the IdentityIQ console.

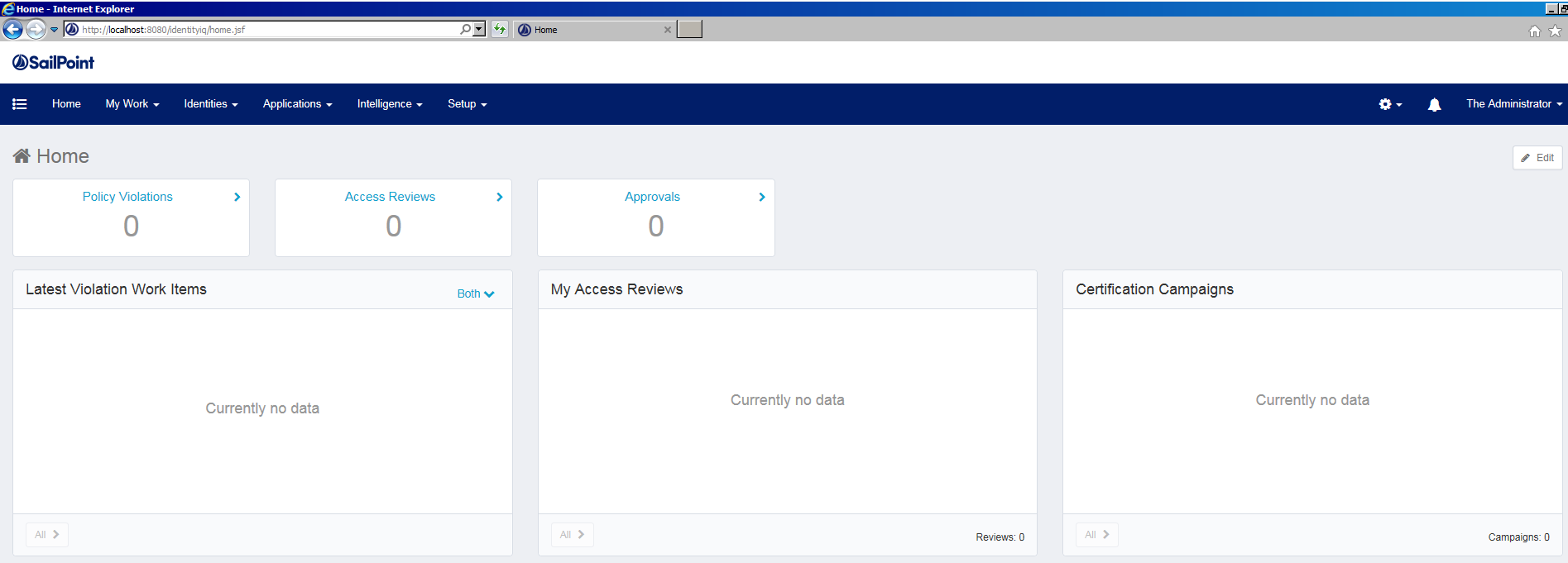
### Launch IdentityIQ

Restart Tomcat. Open a command prompt using “Run as Administrator”, then enter:

\tomcat\bin\Tomcat7w.exe

Then “Stop” and “Start” the Tomcat7 service from the “General” tab in the “Apache Tomcat 7.0 Tomcat7 Properties” window.  The service should take several seconds to complete its restart operation.

Next, test the load of IdentityIQ to make sure it loaded correctly.  To do this you will attempt to connect to the instance of IdentityIQ using a browser.  Open a browser and go to the address [http://localhost:8080/identityiq](https://community.sailpoint.com/localhost:8080/identityiq) .  As this is the first time this server has been logged in to it may take the server a few seconds to respond. You should eventually be greeted with the login splash screen for IdentityIQ.



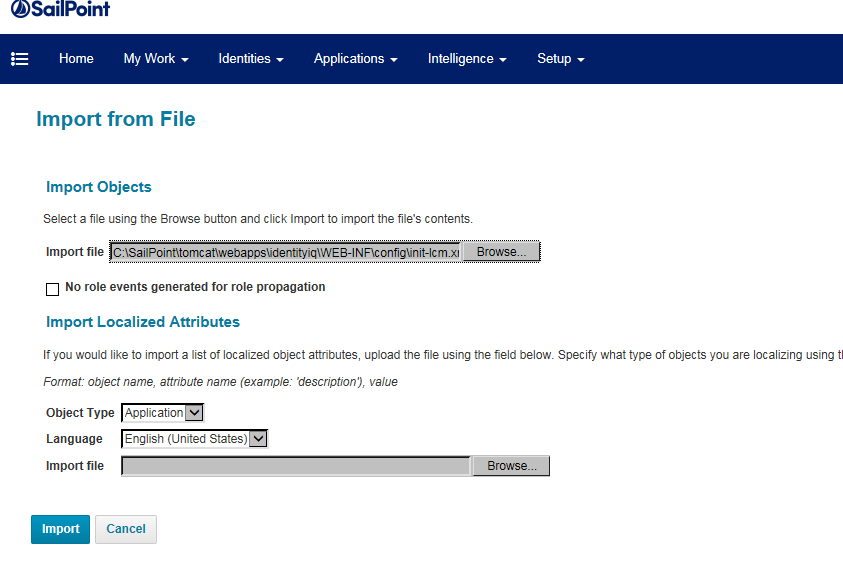
Test logging in with the user “spadmin” and the default password “admin”.  Arriving at a complete dashboard is indication of a successful installation of IdentityIQ.

## Enabling Lifecycle Manager (LCM)

IdentityIQ Lifecycle Manager is a highly flexible solution for managing changes to user access and automating the initiation of provisioning activities within an enterprise. It maps directly to the lifecycle of a user within an organization (joining / moving / leaving) and the core identity business processes which support these lifecycle events (provision / change / de-provision).

To enable IdentityIQ Lifecycle Manager, log in to IdentityIQ using username: “spadmin”, password: “admin”.  Next, click on the “System Setup” tab and select the “Import from File” Page.

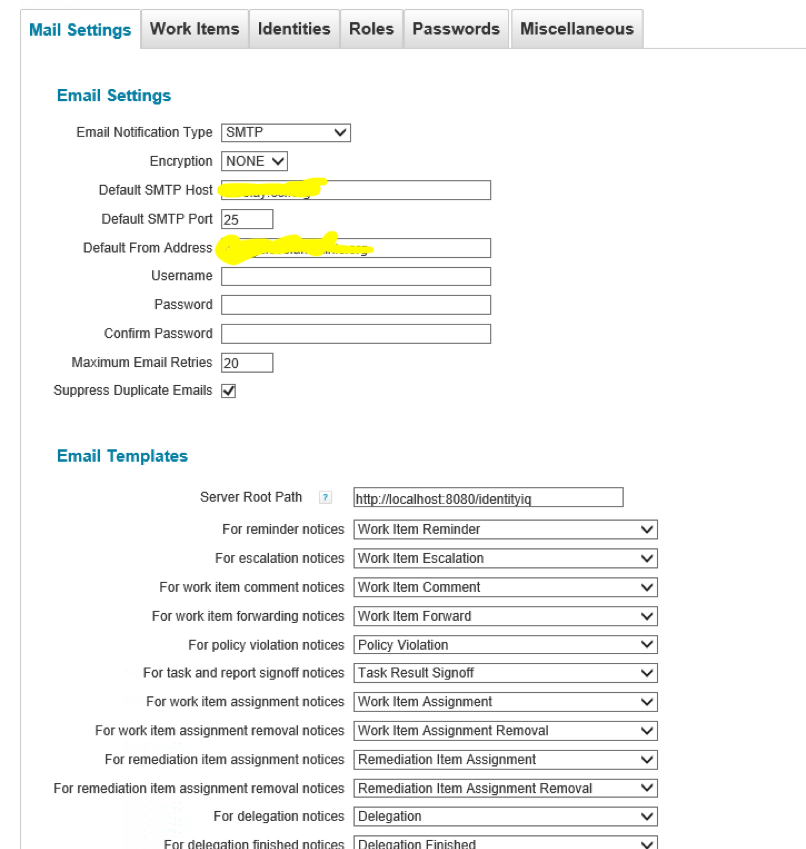
Click “Browse” in the “Import Objects” section of the page, and browse to “\tomcat\webapps\identityiq\WEB-INF\config”.  Select the **init-lcm.xml** file, and click “Import”.



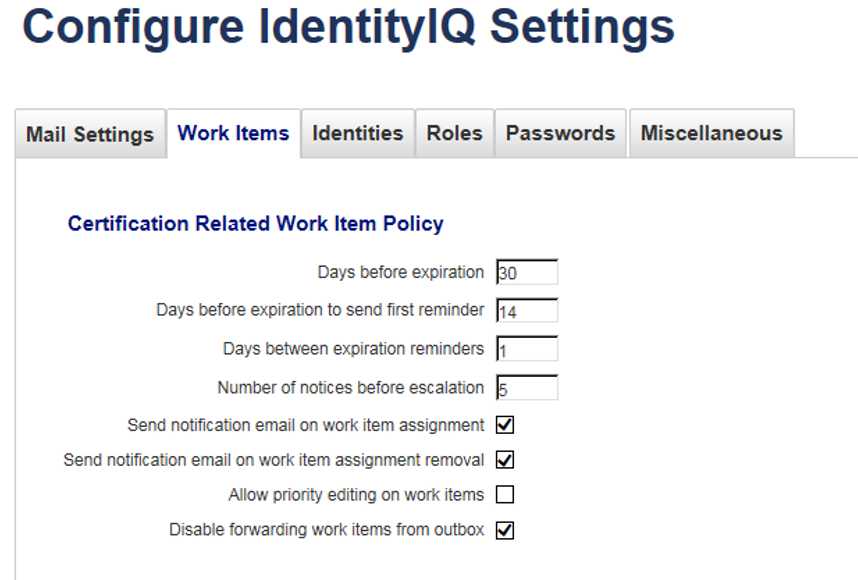
When the import is complete, click “Done”.

## Configuring IdentityIQ

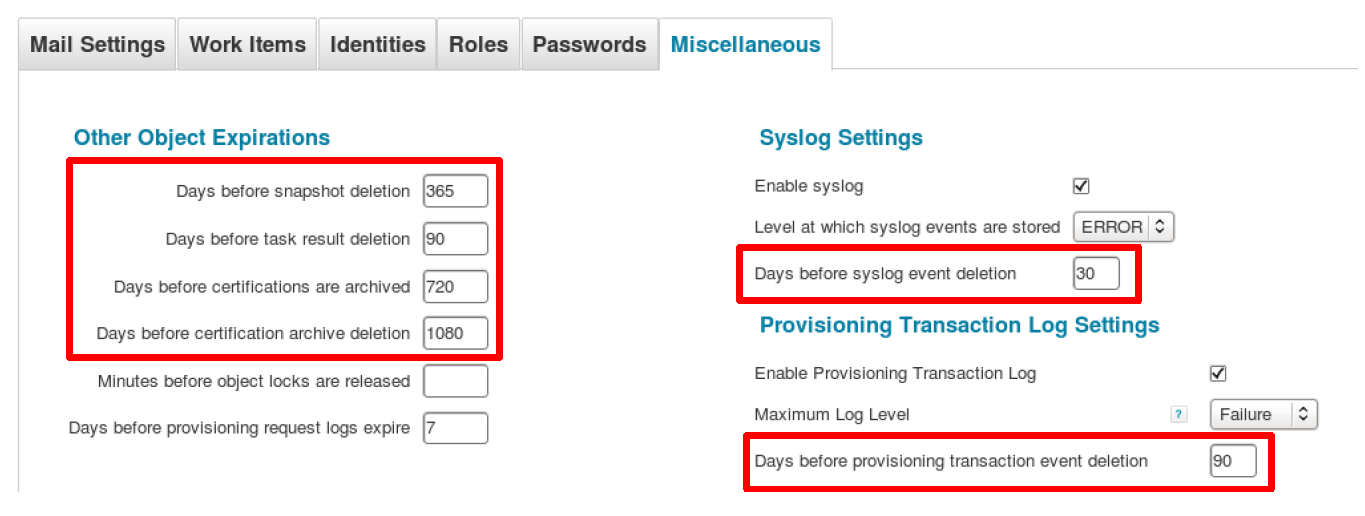
**Setup Email:**



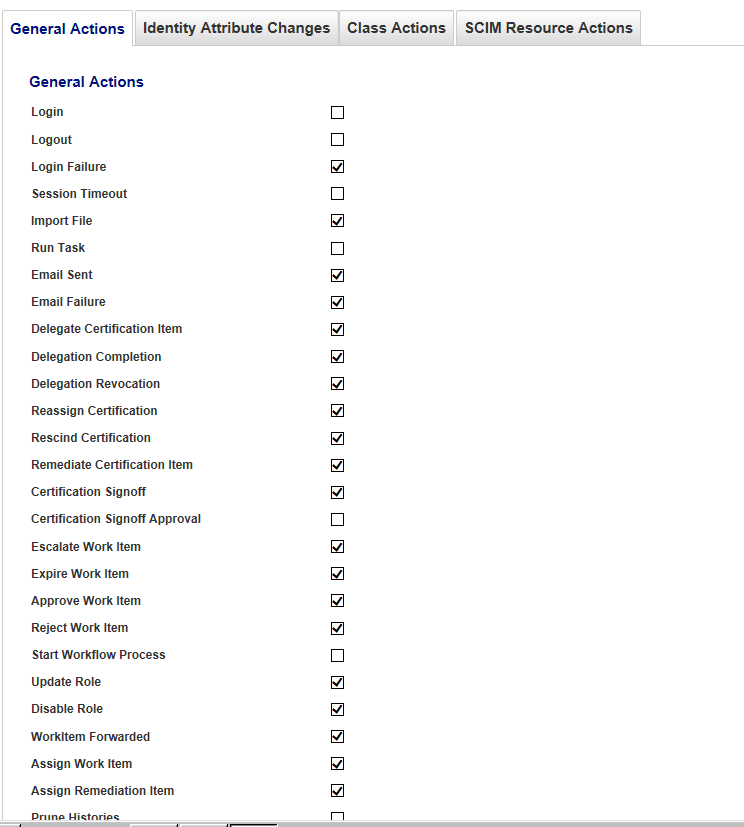
**Setup WorkItems:**

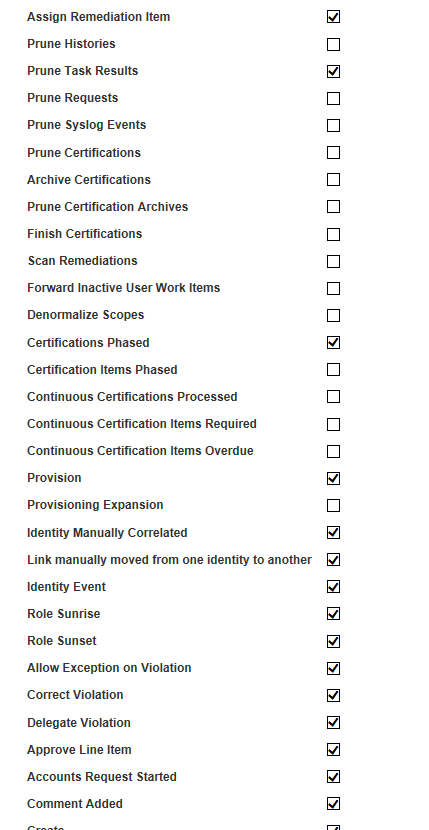


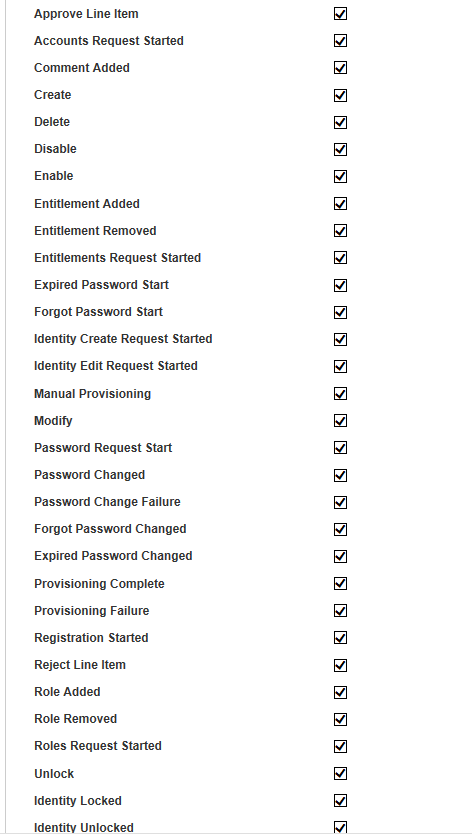
**Miscellaneous**:

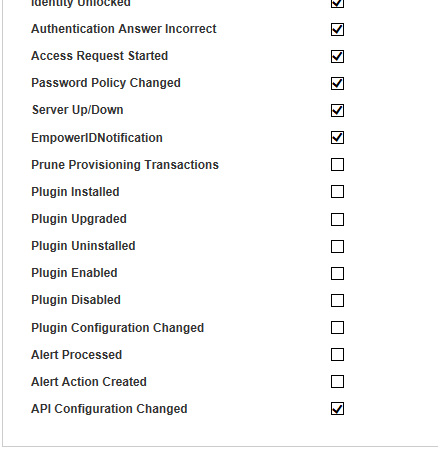


**Audit Config:**









***Configure IdentityIQ Logging***

IdentityIQ uses log4J (a popular Java-based logging package), as its logging component. In this section, we will configure logging by configuring a log4j.properties file:

1. Within the **ImplementerTraining/config** directory, locate the **log4j.properties** file and copy it into **/home/spadmin/tomcat/webapps/identityiq/WEB-INF/classes**

**Note:** This will overwrite the default log4j.properties file with options added for the class environment.

**Copy Options:**

**Option 1:** Use the file browser to copy and paste the file.

**Option 2:** In a command terminal, use the Linux copy command. At the $ command prompt, enter:

**cp log4j.properties /home/spadmin/tomcat/webapps/identityiq/WEB-INF/classes**

2. This sample log configuration file will send all IdentityIQ logging output to the file: **/home/spadmin/logs/iiq\_training\_rolling.log**

3. Finish the configuration process by restarting the Tomcat application server to reload the log4j.properties file.

Options to ***stop*** Tomcat:

**Option 1:** From the desktop, run the “**Stop Tomcat**” shortcut

**Option 2:** Type the following command in a Linux command line terminal:

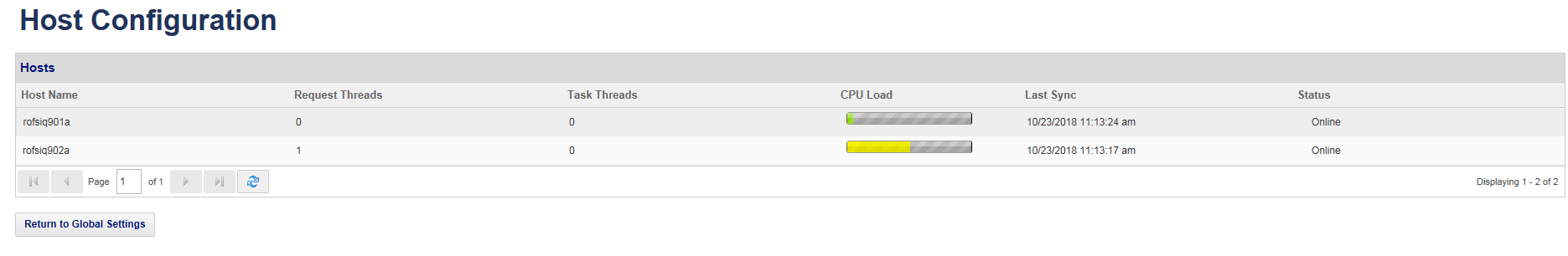
StopTomcat

Options to ***start*** Tomcat:

**Option 1:** From the desktop, run the “**Start Tomcat**” shortcut

**Option 2:** Type the following command in a Linux command line terminal:

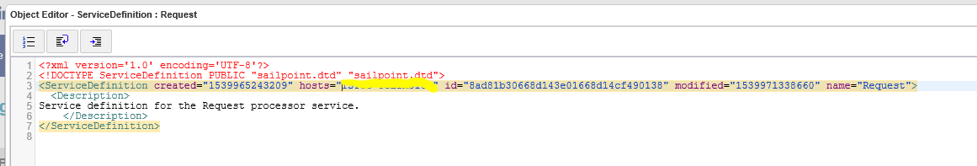
StartTomcat



**Task & UI Servers**

Goto Debug page and type service definition in search box, you will see list of objects, select task from the list & add your hostnames same like as shown in the above image.





## Load Balancer

* LB setup will be done by client
* LB setup will be done for UI servers only and we don’t need task servers unless we are using task server also for requests
* What kind of VIP method should we configure, roundrobin, leastconnection, etc. ?

Manju – Either is fine. I’m ok with roundrobin.

* Should we configure any persistence (sourceip, cookieinstert, etc. ?

Manju – IIQ does not manage session sharing across multiple instances. LB will need to manage persistence, and either sourceip or cookieinsert is required.

Typically, sourceip is fine if the internal CLIENT users are coming from different IP addresses (not thorugh proxy) and they do not change in their network. Otherwise, go with cookieinsert.

* Is any specific monitoring needed to test VIP health or only the standard ping testing ?

Manju – We created a html page and added it to iiq project folder. If html is accessible, then iiq is deployed. We can use below URL’s that will check if tomcat application server is running and if the IIQ application is deployed.

<http://yourhost:8080/identityiq/pingLB.html>

We have created a simple html page in the SSB project under web folder as shown below. When we explode the SSB project, this html will be created under root /identityiq folder and confirms us that tomcat server is running and IIQ SSB project is deployed.



## Setup SSL

* Get the .pfx file to be installed on your server and save it to the tomcat server location **\tomcat\conf**
* Update the SSL section of the server.xml configuration file with the following information:

<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443 for CLIENT IIQ-->

<Connector port="443" maxHttpHeaderSize="8192" maxThreads="150"

minSpareThreads="25" maxSpareThreads="75" enableLookups="false"

disableUploadTimeout="true" acceptCount="100" scheme="https"

secure="true" SSLEnabled="true" clientAuth="false"

sslProtocol="TLS" keystoreFile="conf/samplecert.pfx"

keystorePass="password" keystoreType="PKCS12"/>

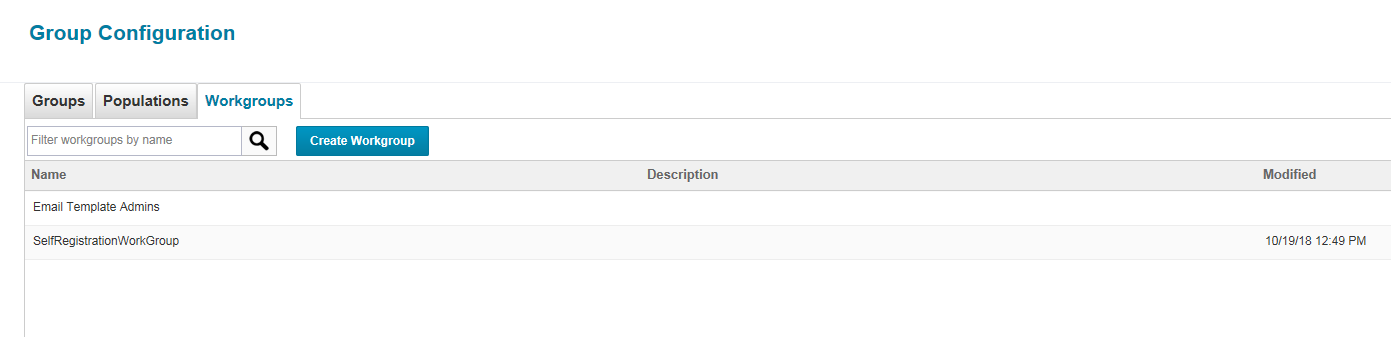
* Save the server.xml file
* Restart Tomcat for your new certificate to take effect.

## System Administrator Capability

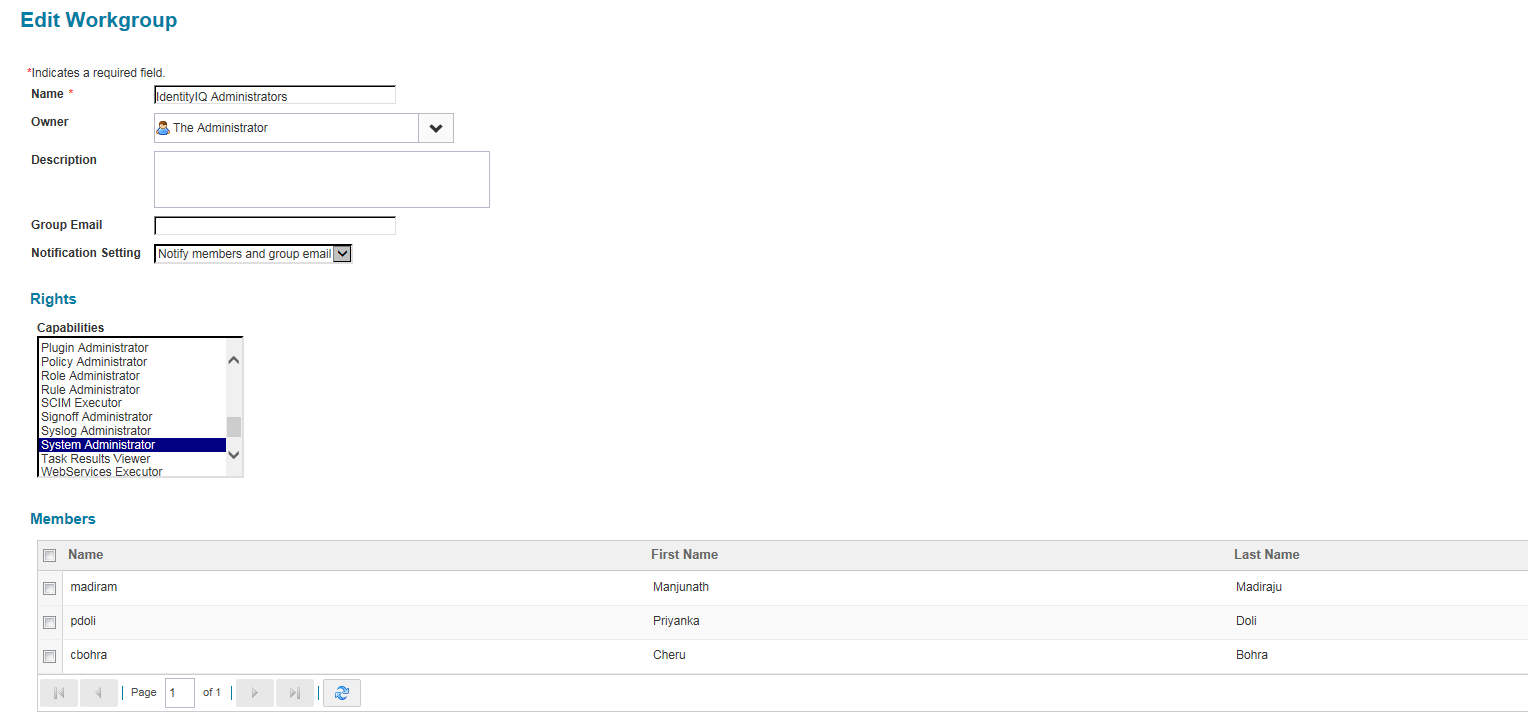
## The spadmin account is the most powerful account of the system, like root on a Unix system or Administrator on Windows. A common recommendation for securing a Windows installation is to create an alternate administrative account and disable the original. This recommendation is useful for IdentityIQ as well. In case an alternate spadmin account is created, make sure to also make this account protected by setting the protected attribute on the object. This will ensure that the account is never accidentally deleted:<Identity name="altadmin" password="\*\*\*\*" protected="true">

* Created an identity for users from debug mode
* Create a workgroup in IdentityIQ and assign the SystemAdministrator capability to that workgroup. Then all the identities that need the SystemAdministrator capability can be added to the workgroup as members.

Navigate to setup 🡪 groups



In workgroup, create Admin group.



## Encryption

To prevent easy decryption of accidentally leaked, or illegally retrieved sensitive data, it is recommended to enable the keystore feature of IdentityIQ. Without the keystore, a default key is used to encrypt all sensitive data within IdentityIQ. This means that even though a password is encrypted, it can easily be decrypted on any other instance of IdentityIQ. With the keystore feature, a system specific encryption key is created, that is required to decrypt any sensitive data.

The keystore itself is stored in WEB-INF/classes/iiq.dat with an accompanying configuration file WEB-INF/classes/iiq.cfg. Neither of these files is human readable.

**Key Creation**

In order to create or manage the keystore, start the IdentityIQ command line tool with the 'keystore' command, from the WEB-INF/bin folder:

On Windows:

C:> iiq.bat keystore

In order to see whether a site specific key already exists, use the list command.

> list

Listing contents for keystore [/var/tomcat/webapps/identityiq/WEB-INF/classes/iiq.dat].

KeyAlias Algorithm Format Object

No keys found.

>

In this case the file is empty. To create a key, use the addKey command:

> addKey

Generate a new encryption key (y/n)?

y

Generating a new encryption key for keystore [/var/tomcat/webapps/identityiq/WEB-INF/classes/iiq.dat].

New encrpytion key successfully saved to keystore.

All application servers must be restarted for changes to take effect.

>

The tool asks you whether you'd like to generate a new encryption key. Enter 'y' and press enter to continue. If the keystore file does not exist, it will be created and a new, randomly generated key will be added.

The list command will now show the newly created key:

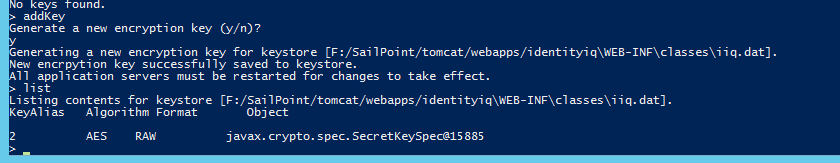
> list

Listing contents for keystore [/var/tomcat/webapps/identityiq/WEB-INF/classes/iiq.dat].

KeyAlias Algorithm Format Object

2 AES RAW javax.crypto.spec.SecretKeySpec@fffe81cd

>



Exit the tool with the exit command:

> exit

Now restart your application server. After restarting the application server any newly set password will be encrypted using the new encryption key. Without the files iiq.dat and iiq.cfg, passwords cannot be decrypted by IdentityIQ.

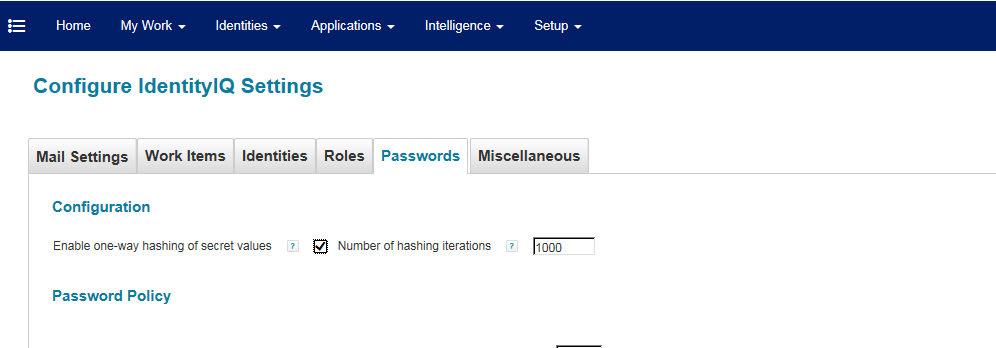
If you run more than one instance of IdentityIQ, these files iiq.dat and iiq.cfg need to be placed in the WEB-INF/classes folder of each instance, or in the location specified in iiq.properties.

**Note:**Make sure to store copies of the files iiq.dat and iiq.cfg in a safe place, so they can be restored after upgrading or reinstalling IdentityIQ.

**Note:**Make sure that the file permissions are set to allow access only by the application server that runs IdentityIQ.

## Re-Encrypt Passwords

A new feature was introduced for one-way hashing (IdentityIQ uses SHA-256) of identity password, password history and authentication question answers. This can be configured in the System Setup (Identity Configuration) as shown below:



Now, the new encryption key will be used for newly encrypted passwords, but existing passwords can still be decrypted using the default method on any system. To prevent that, existing passwords need to be re-encrypted. To do so, you need to create a new task of type “Encrypted Data Synchronization Task”.

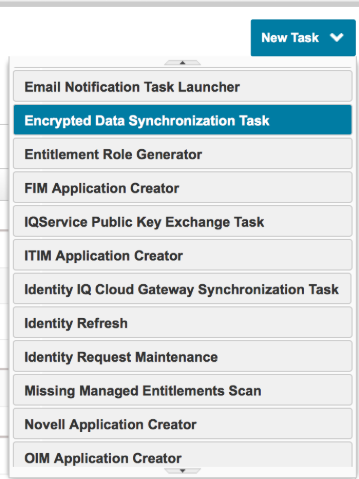
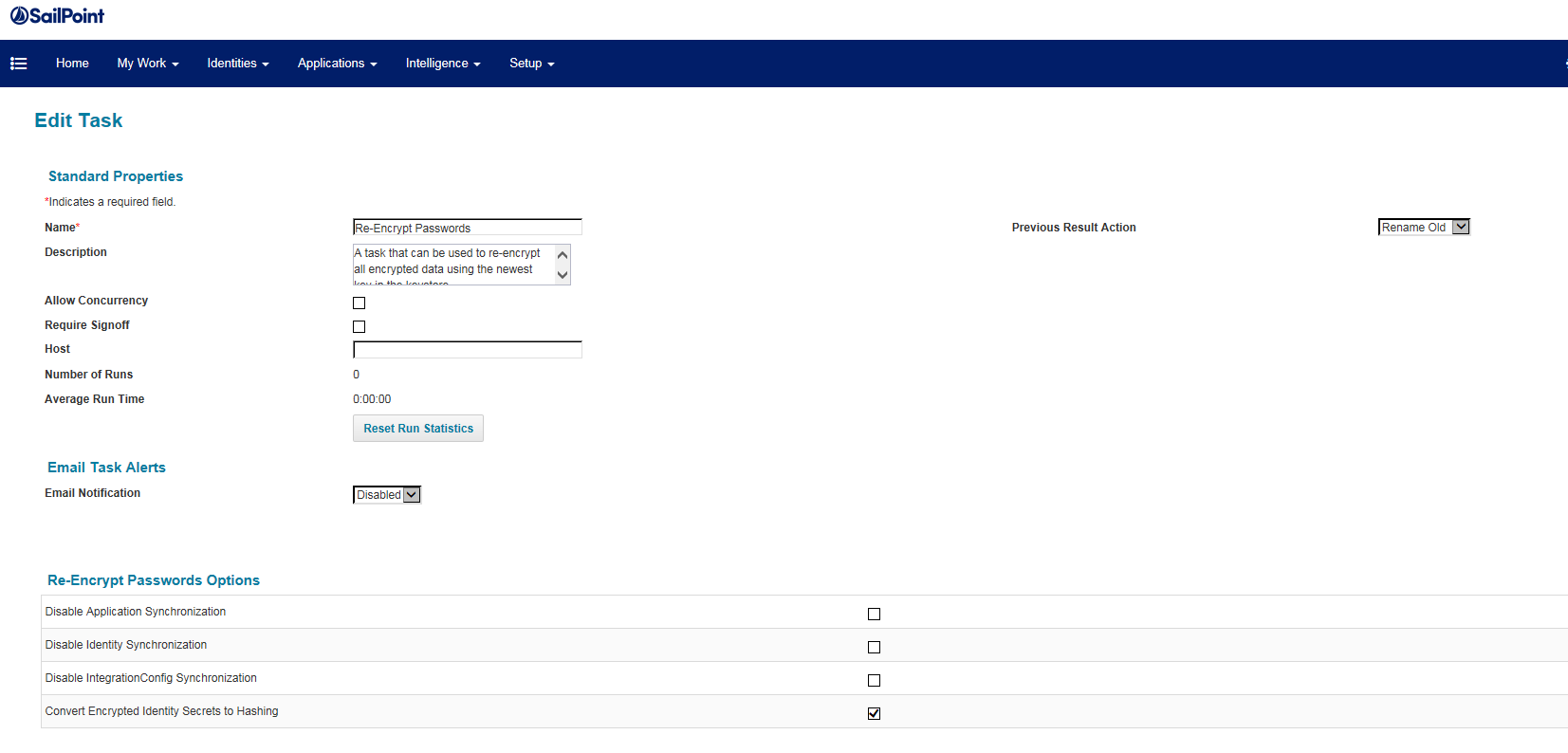


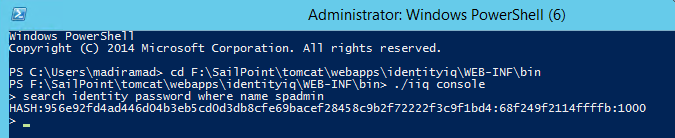
Illustration 1: Create Encrypted Data Synchronization Task

Give the task a name and optionally exclude types (applications, identities or integration configurations) from processing. Click “Save and Execute” to immediately run the task.



After the task has completed, all (selected) encrypted data will be changed. While a password encrypted with the default key is prefixed with “1:”, items encrypted with the new encryption key are prefixed with “2:” or another number if multiple encryption keys are stored.

When looking up The Administrator's password in the console, this will look like:



Or in the debug interface:

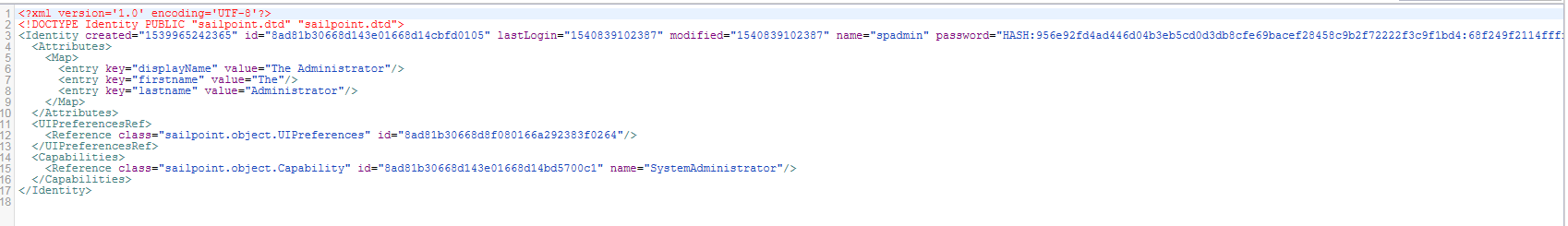


Illustration 2: Re-Encrypted Identity Password in Debug Interface

Using the Different Encryption Keys

After a new key has been added to the keystore, this will be the default encryption key. Everything encrypted inside IdentityIQ from that moment on will use this key. For example:

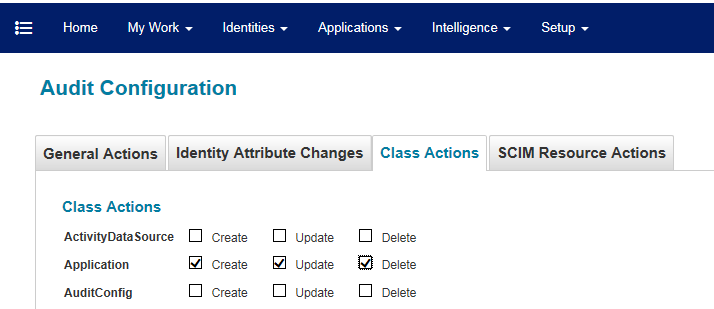
1. $ ./iiq console
2. > encrypt spadmin password
3. 2:bt7YJA6iovzF5Uu6RIjueg==
4. >

 In the debug mode, update the password for spadmin to the encrypted value from above action. Once you save, it will automatically convert it to hash format.



## Object Level Auditing

Object level auditing can be configured to audit changes to the objects shown in the figure below. Any changes to the following objects (if checked) will trigger creation of audit events. For now, it’s just set for Application.

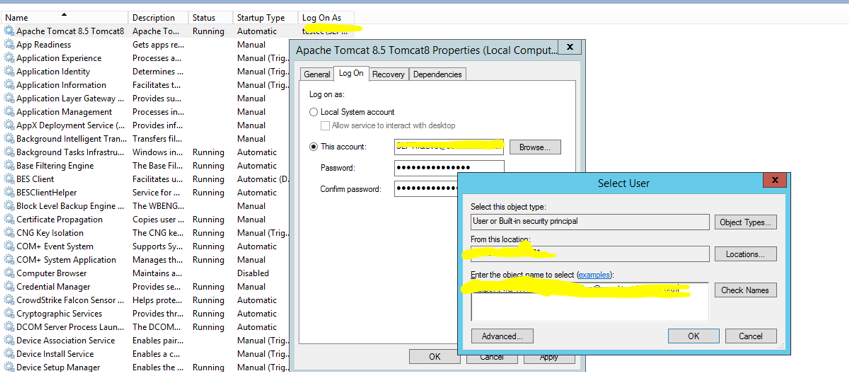


## Domain Account for secure operating system

Services on Windows often run as the system user. The system user has all permissions on the local system, which is a potential security risk. It is therefore recommended that you create a non-privileged local or domain account to run the service for the application server that serves IdentityIQ.

Using a domain account can be useful when the service account needs to access remote file shares or other services using Active Directory credentials (Kerberos tickets) to gather data. If no access to other systems through a network account is required, or other means are available, it is preferred to use a local system account. This will prevent any compromised web applications running on the application server from accessing other systems using acquired tickets or credentials.

* Right click on Apache Tomcat service and navigate to Logon tab
* Click browse 🡪 Enter the service account name create 🡪 Check Names 🡪 Select user and and click OK



* Enter the password and then click Apply
* Restart the tomcat server

## Configuring IdentityIQ to communicate with Active Directory over SSL

Follow the instructions in the below Sailpoint

<https://community.sailpoint.com/docs/DOC-4463>

## Generate IQService Server Certificate

If you have IQService WITH TLS enabled for AD application, you need to generate the IQService certificate and import it into IIQ servers cacerts.

To setup TLS connection between IIQ server and IQService server. Below are the details required:

**FQDN: IQService server name**

**Certificate Configuration Steps:**

Ensure that the following mentioned configurations are performed on IQService computer:

* The X.509 certificate is located in the local computer's personal certificate store (as shown in Figure 1 below (**Console Root ==>Certificates (Local Computer) ==> Personal ==> Certificates**)).
* A private key that matches the certificate is present in the local computer's store and is correctly associated with the certificate (in Figure 1 it is mentioned as **Green - Certificate with corresponding private key**)

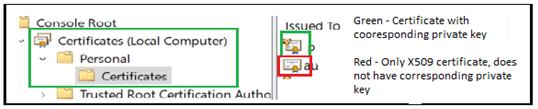


        Figure 1 - Certificate MMC (Local Computer)

* The Enhanced Key Usage extension includes the Server Authentication (1.3.6.1.5.5.7.3.1) object identifier.
* The IQService computer’s fully qualified domain name (FQDN)) (for example, [**iqservcie.test.com**](http://iqservcie.test.com)) must appear in:
  + The Common Name (CN) in the Subject field

               and

* + DNS entry in the Subject Alternative Name extension
* The **X.509** certificate must be trusted by IQService computer (as shown in Figure 3 below).

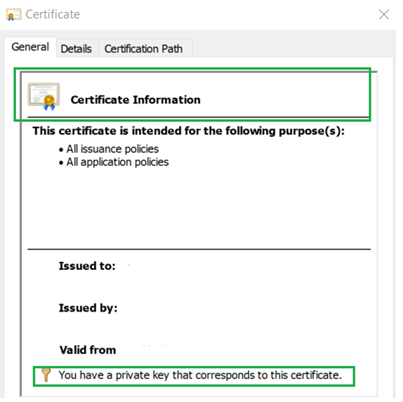
Trust is established by configuring the IQService computer to trust the CA Root certificate which is issued. For more information on configuring the CA Root Certificate on IdentityIQ, see [CA Root Certificate Configuration on IdentityIQ](https://community.sailpoint.com/docs/DOC-13018#jive_content_id_Root_CA_Certificate_configuration_on_IdentityIQ_).

* + **X.509 certificate (Non trusted)**



             Figure 2 - X.509 certificate (Non trusted)

* + **X.509 certificate (Trusted)**



## Configuring IdentityIQ to communicate with Active Directory over SSL

1. Login to all UI and task servers, navigate to **C:\Program Files\Java\jre1.8.0\_181\lib\security.** Execute the following command to import certs in default keystore.

|  |
| --- |
| keytool -import -alias ClientRoot -keystore cacerts.jks -file C:\CLIENTNAMERootCA\Root.cer |
| keytool -import -alias ClientServerIssuer01 -keystore cacerts.jks -file C:\CLIENTNAMERootCA\Issuer01.cer |
| keytool -import -alias ClientManagedIssuer01 -keystore cacerts.jks -file C:\CLIENTNAMERootCA\ManagedIssuer01.cer |
| keytool -import -alias ClientServerIssuer02 -keystore cacerts.jks -file C:\CLIENTNAMERootCA\ServerIssuer02.cer |
| keytool -import -alias iqservicecert -keystore cacerts.jks -file C:\CLIENTNAMERootCA\iqservice.cer |

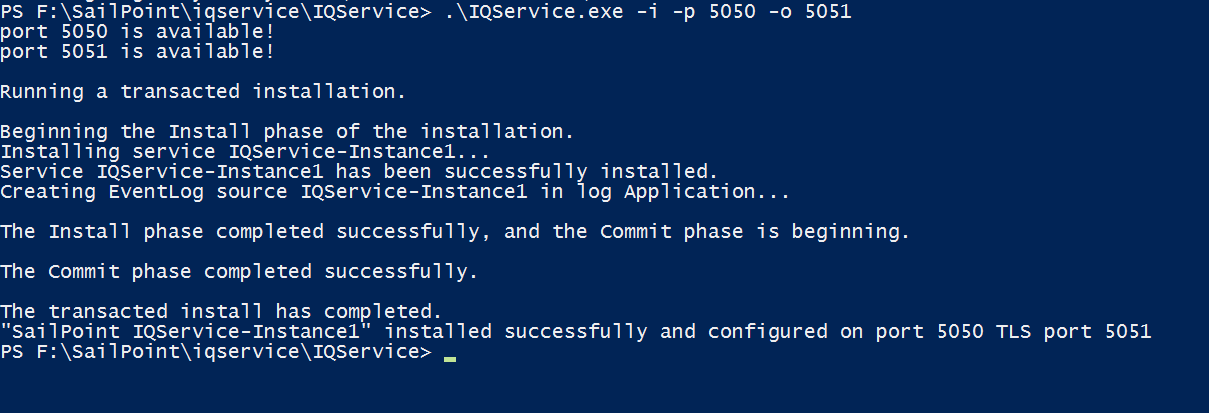
**Configuring IQService for communicating with Active Directory over SSL**

* Install JDK on the IQService server and set JAVA\_HOME and Path.
* Double-click the .cer file.
* Click Install Certificate then click Next.
* Choose Place all certificates in the following store and click Browse...
* Select Show physical stores.
* Expand Trusted Root Certification Authorities and select Local Computer.
* Click OK.
* Click Next, then Finish.

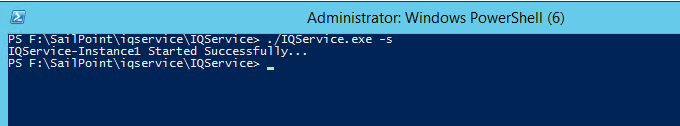
## Install and Register the IQService for Use with Windows

Install the IQService in a location that is running Windows. To install and register the IQService, do the following:

1. Create a directory in which to download the service. For example, \iqservice
2. Copy the IQService.zip file from the IdentityIQ installation into the new directory. The IQService.zip file is located in *identityiq\_home*/WEB-INF/bin/win where *identityiq\_home* is the directory in which you expanded the identityiq.war file.
3. Expand the IQService.zip.
4. Open command prompt and navigate to the path where you exported IQ Service “\iqservice\IQService”
5. Run **IQService.exe -i -p 5050 -o 5051** to install a Windows service named IQService.



1. Start the service from Windows Services Applet or from the command line by running **IQService.exe -s**



If we need to enable log tracing for IQ Service then enable debugging by invoking "IQService.exe -l 3" where value 3 is full verbose and 0 is off the trace. and then restarting IQService with "IQService.exe -t" .

**Install additional powershell tools for Active Directory on the IQService server**

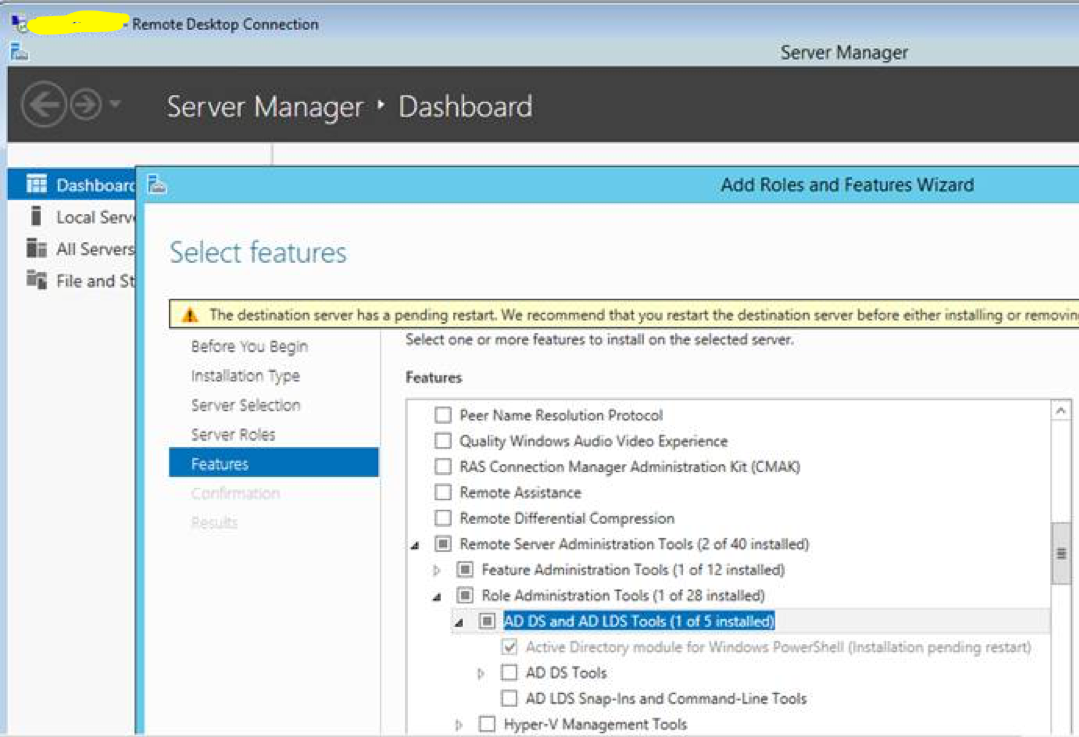
This is required to avoid few connector exceptions like below:

"**The term 'Get-ADDomainController' is not recognized as the name of a cmdlet, function, script file, or operable program."**

Steps to  install:

1) Server manager > Add Roles and Features > click Next at "Before you begin" step > click Next at "Installation Type" step > click Next at "Server Selection" step  > click Next at "Server Roles" step.

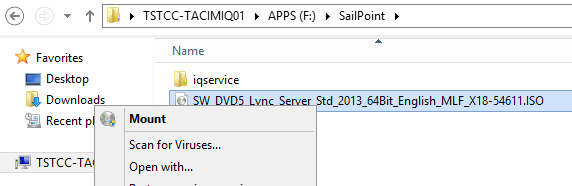
2) Check "Active Directory module for Windows PowerShell" as below :



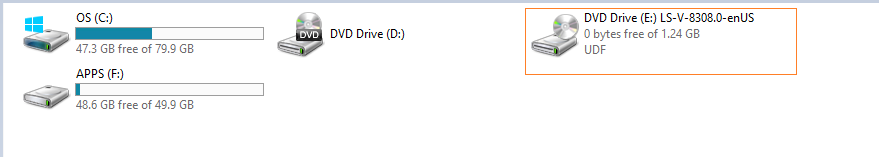
3) Continue installation and restart IQService.

## Install Lync Administrative Tools if required

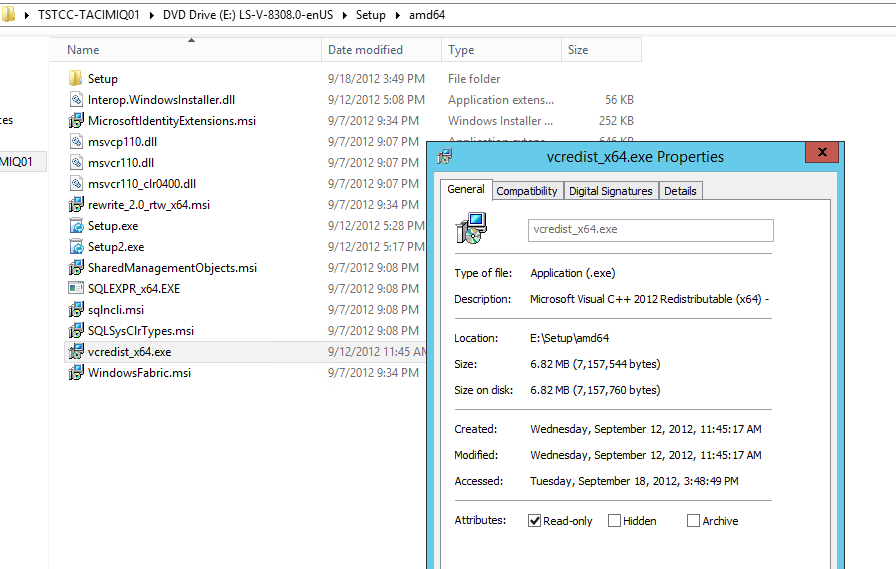
* + - Client Will provide the Lync ISO file.
    - Right click on the ISO file and mount it

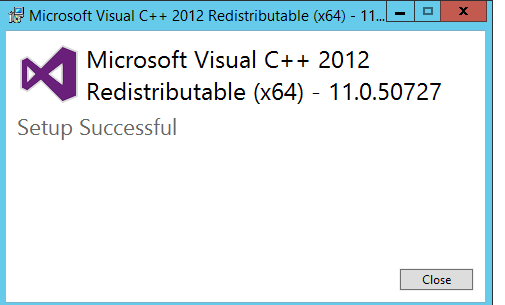


* + - In My Computer, we can see a new drive. Open the drive to access the installation files.

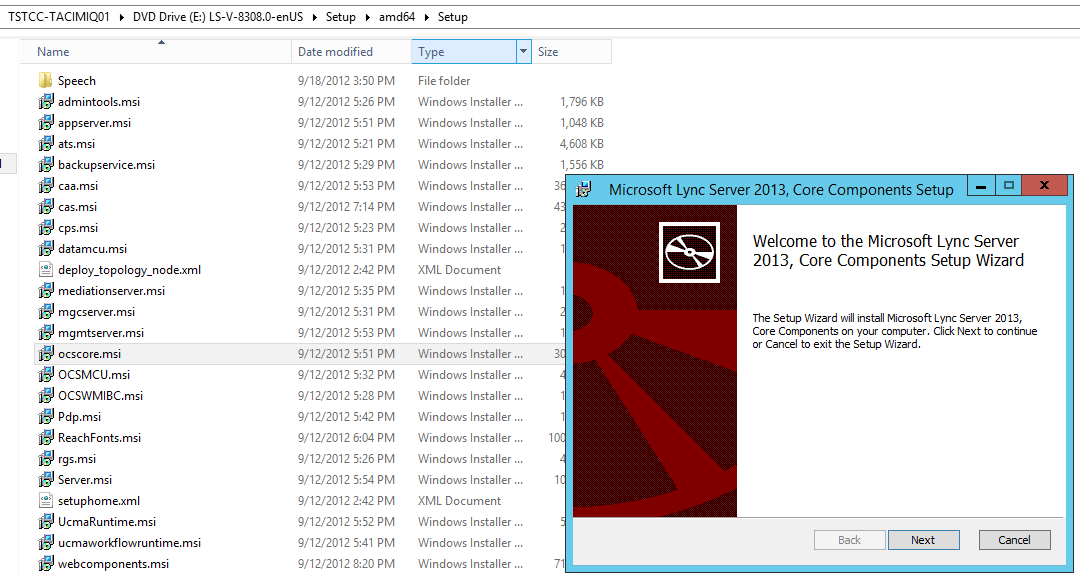


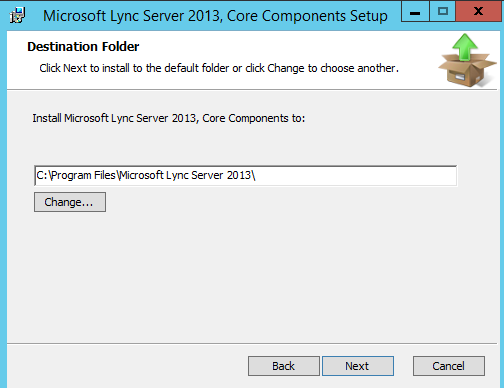
* + - If you see the below error around VC++ version, you have to have the exact version as shown in the next slides. The 11.0.6 version won’t help in fixing this error. We can find the 11.0.50727 version only with Lync Server 2013 image. Right click on the file and install it.



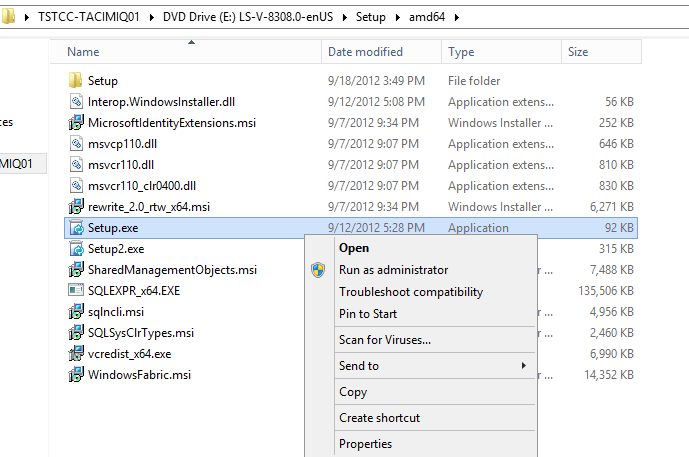


* + - Install ocscore.msi before installing Lync 2013 admin tools

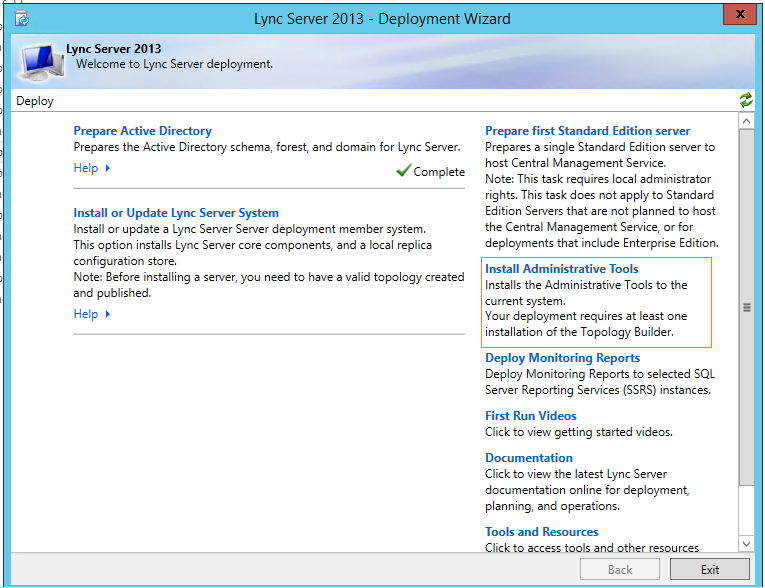




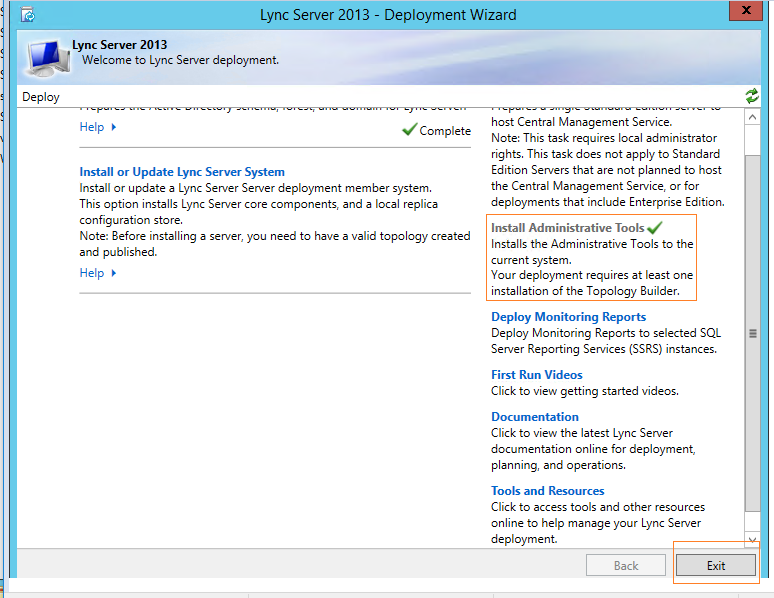
1. **Lync Server 2013 administrative tools Instalaltion:**
2. Log on as a local administrator (minimum requirement) to the computer where you want to install the administrative tools.
3. Locate the installation media on your computer, and then double-click \Setup\amd64\Setup.exe.



1. On the **Microsoft Lync Server 2013 – Deployment Wizard** page, click **Install Administrator Tools**.

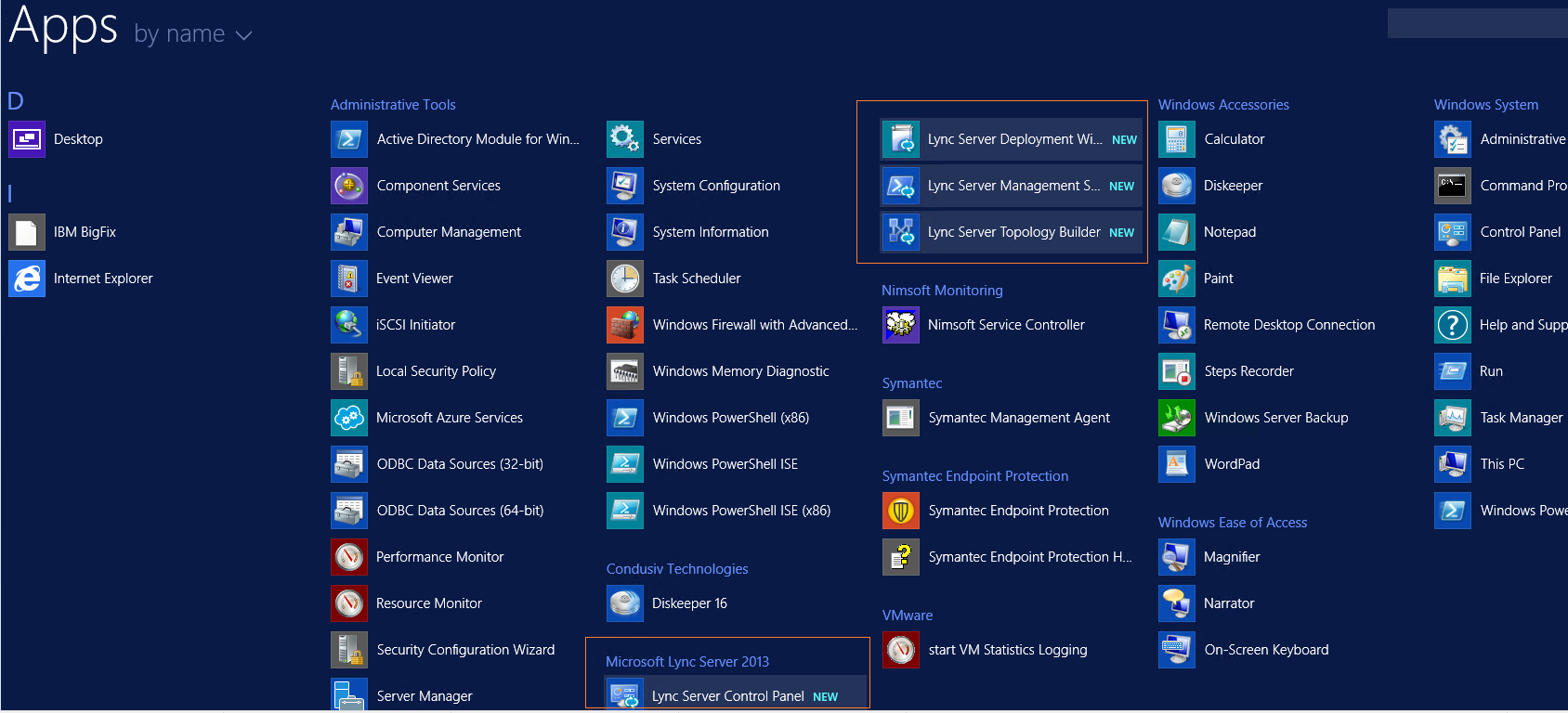


1. When the installation successfully completes, click **Exit**



1. **Post installation verification:**

Post install, you should be able to see the Exchange/Lync tools as highlighted below:

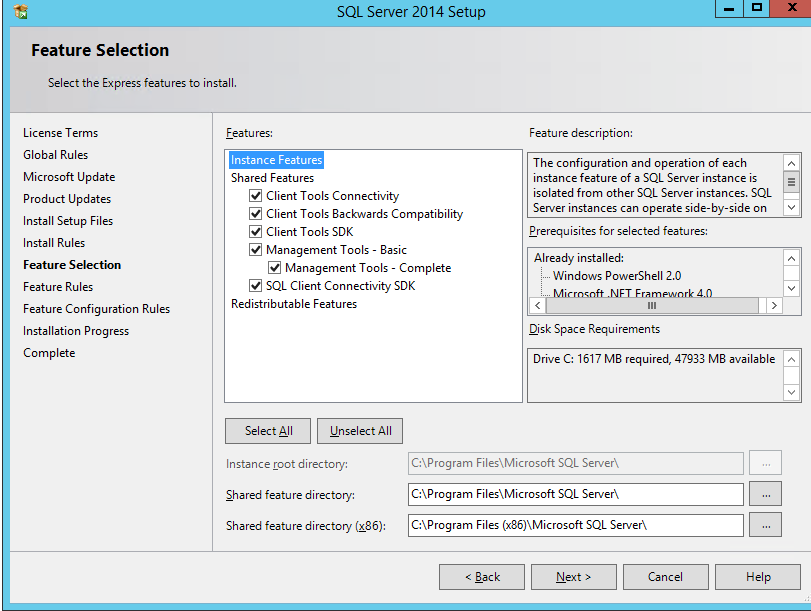


1. Copy Lync modules **Lync** and **LyncOnlineConnector** from **C:\Program Files\Common Files\Microsoft Lync Server 2013\Modules\** to **C:\Windows\System32\WindowsPowerShell\v1.0\Modules**

## 

## Microsoft SQL Server Client

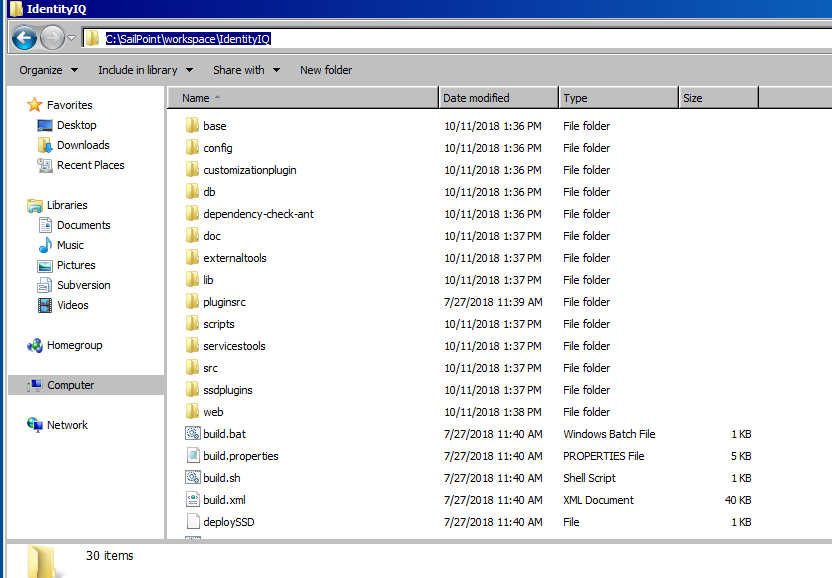
* + - Download SQLManagementStudio\_x64\_ENU.exe from Microsoft downloads
    - Right click and run as administrator



## SSD Setup

The Services Standard Deployment (SSD) is a set of tools and artifacts designed to increase the ease of installation and add a level of standardization across IdentityIQ deployments. These artifacts and tools are used by the SailPoint Services team on projects deploying IdentityIQ.

* + 1. Download and unpack the latest SSD zip file ssd\_v6.zip to workspace folder C:\SailPoint\workspace\IdentityIQ



* + 1. Open a terminal or command prompt and cd to the root folder of your unzipped SSD project. Validate the contents:

• /config/SSF\_Features

• /config/SSF\_Frameworks

• /config/SSF\_Tools

• build.properties

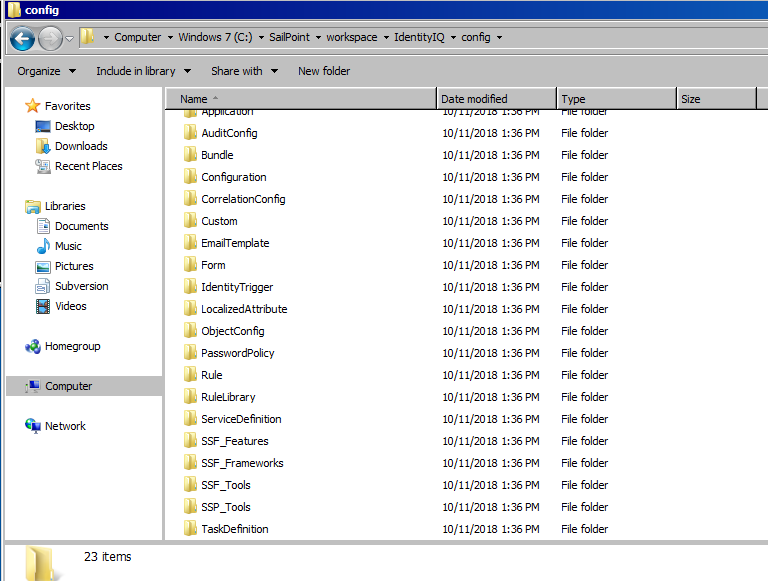
• ssf.ignorefiles.properties

• ssf.target.properties

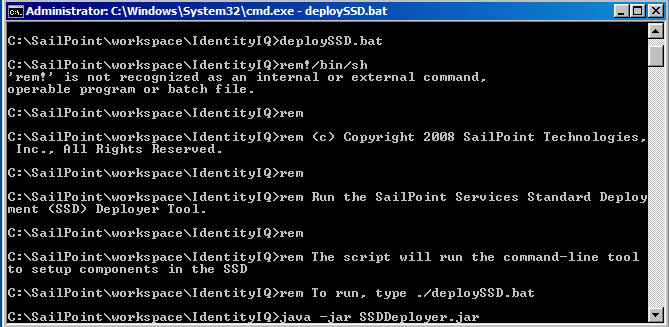
• SSDDeployer.jar

• deploySSD

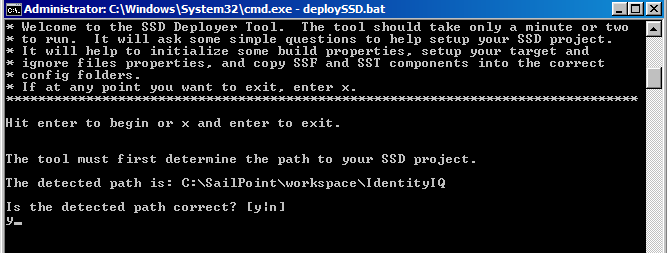
• deploySSD.bat



* + 1. The following details the prompts and inputs required, most prompts simply requiring a “y” or an “n” response. The input process includes the following steps:
* Start the tool (using the deploySSD or deploySSD.bat script at the root of the SSD)
* Validate or enter the SSD path
* Enter environments
* Enable individual components
* Review summary and start the deployment

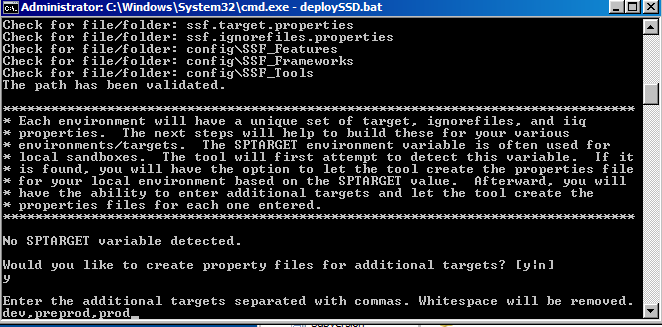


Hit enter

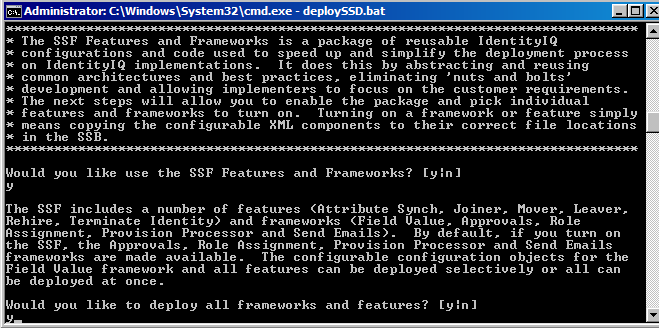


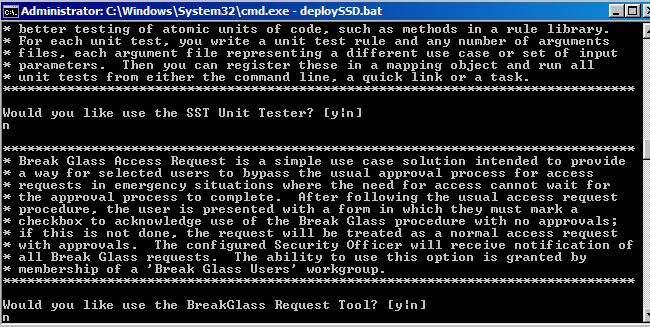
Validate path and Enter “Y”.

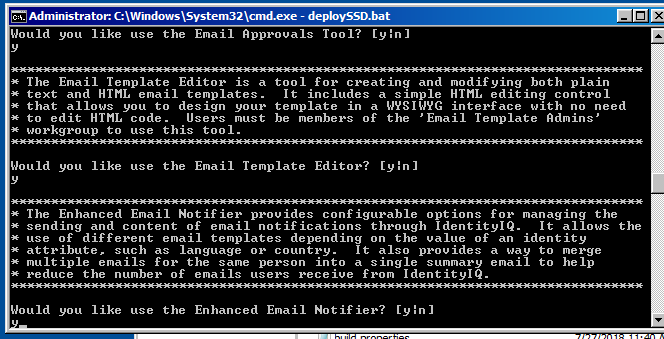
Select Y to create property files for additional targets and Hit enter. Specify Dev, Preprod and Prod as targets.

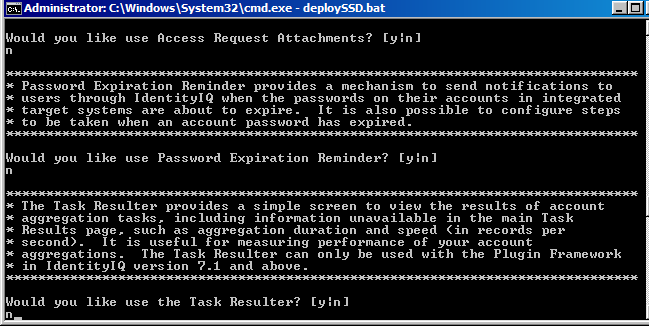


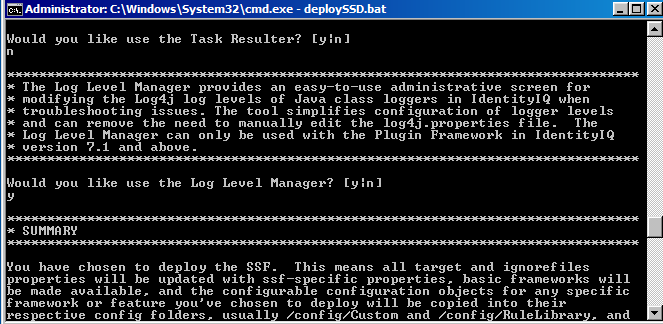
Hit enter. Select Y for SSF Features and Frameworks

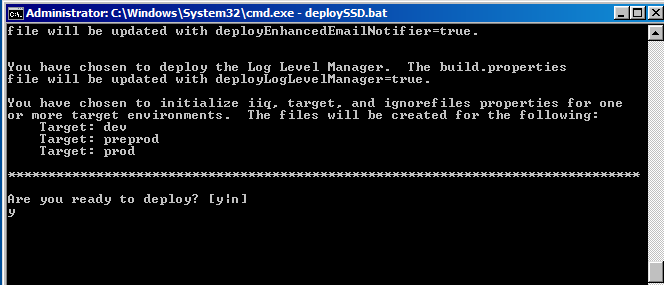


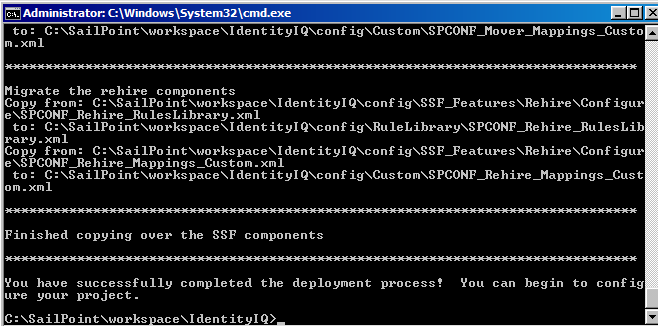












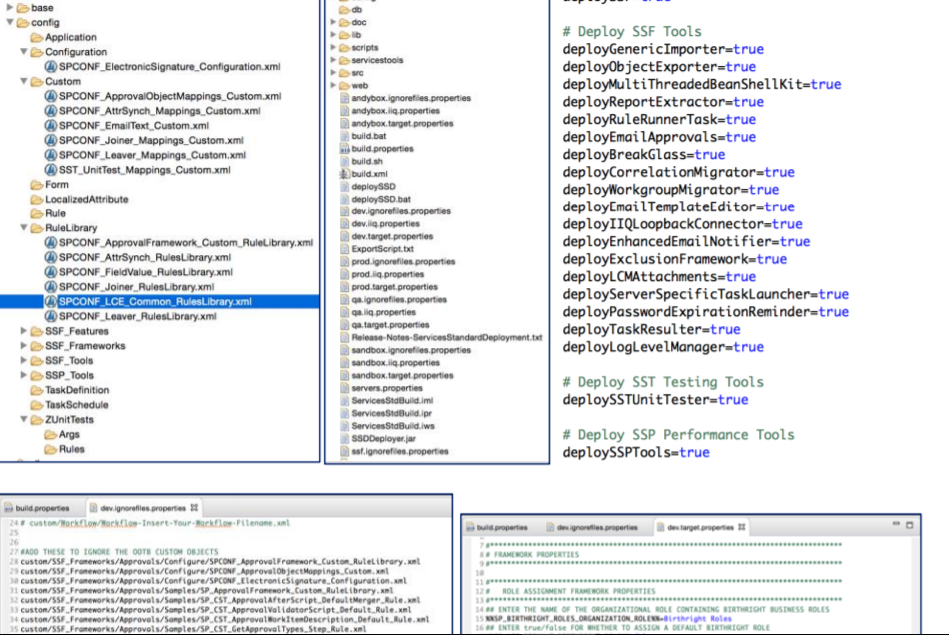
* + 1. The following shows the updated package after running the SSD Deployment Tool. The following changes have been made:

• The Configuration, Custom, RuleLibrary folders under /config have been populated (if the SSF Features and Frameworks were selected for deployment)

• For each host environment entered, a new iiq.properties, target.properties, and ignorefiles.properties file has been created

• The target.properties and ignorefiles.properties have been copied into each new respective file for each component selected

• The build.properties has been updated, setting the deploy options for each component to true or false depending on which components were selected for deployment.



**SSB Configuration:**

* + - Copy Identity IQ zip file to /base/ga directory.

/base/ga/identityiq-7.2.zip

* + - Copy IdentityIQ7.2Patch2 file to /base/patch directory.

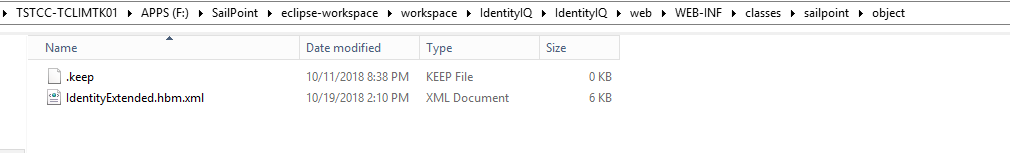
/base/patch/identityiq-7.2p2.jar

* + - First, get your updated JDBC driver and place it in the web/WEB-INF/lib area of the SSB project. For SQL Server, this is **sqljdbc42.jar**.
    - Added the IdentityExtended.hbm.xml file with all the custom attributes in below format to “\eclipse-workspace\workspace\IdentityIQ\IdentityIQ\web\WEB-INF\classes\sailpoint\object”

<property name="cchsid" type="string" length="450"

access="sailpoint.persistence.ExtendedPropertyAccessor"

index="spt\_identity\_cchs\_id\_ci"/>



* + - Configuring the build.properties file The **build.properties** file is a crucial configuration file that specifies many important configuration arguments, like the version of IdentityIQ you are running, the Customer name, and the path to your IdentityIQ installation. Without this information, the build cannot run successfully. Now configure the **build.properties** file found in the <SSB Install directory>. Use your favorite text editor to edit this file.



* + - Setting up environment-specific IIQ properties files
      * The **iiq.properties** file contains properties used by IdentityIQ for connecting to and interacting with its own database. Your build environment can specify different iiq.properties files for the build to use for deploying to each target environment.
      * Copy the **iiq.properties** to the **<SSB .iiq.properties>** file.
        + Copy WEB-INF/classes/iiq.properties to dev.iiq.properties
        + Copy WEB-INF/classes/iiq.properties to preprod.iiq.properties
        + Copy WEB-INF/classes/iiq.properties to prod.iiq.properties
      * Update **<environment>.iiq.properties** files for Dev, Preprod and Prod with environment specific values like database details.

dev.iiq.properties

preprod.iiq.properties

prod.iiq.properties

* + - * Example dev.iiq.properties file below



* + - Configuring target.properties files
      * It is important to configure environment-specific properties files that the SSB can use to do token string replacements in the objects during the build process. The SSB will automatically look for tokenized strings in your custom configuration XML and substitute the appropriate values per environment. A **target.properties** file should be created for each environment, containing key/value pairs for token substitution during build time. The name of each target.properties file should be in the format **<environment>.target.properties**.

dev.target.properties

preprod.target.properties

prod.target.properties

* + - * Each file is just a list of key/value pairs. The build’s convention is that the keys follow a %%KEYNAME%% pattern.

%%AD\_IQSERVICE\_HOST%%=iqservicehost.example.com %%AD\_IQSERVICE\_PORT%%=5051 %%AD\_PROXY\_USER%%=productionADuser %%AD\_PROXY\_PASSWORD%%=2:omj3oouHSFb7dIPItTjNIgBCeZjxP+Vr9TewSXIIbxs

* + - * Sample dev.target.properties file attached



* + - Configuring ignorefiles.properties files
      * The build process can be configured to define certain XML files that should be skipped during the import for a specific environment. Implement this by creating a text file for each environment where you want to skip the import of specific files. A sandbox example file is provided with the build (sandbox.ignorefiles.properties). You can copy, paste, and rename this file for all your environments as a template to get started.

dev.ignorefiles.properties

preprod.ignorefiles.properties

prod.ignorefiles.properties

* + - * Attached dev.ignorefiles.properties below

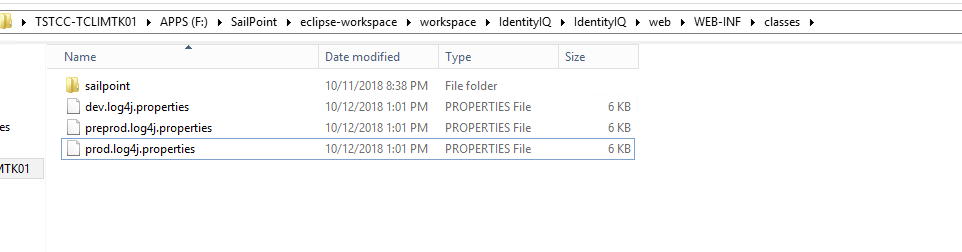


* + - Configuring log4j.properties files
      * Copy log4j.properties file from **\tomcat\webapps\identityiq\WEB-INF\classes** to **\eclipse-workspace\workspace\IdentityIQ\IdentityIQ\web\WEB-INF\classes** as 3 environment specific log4j.properties files.

dev.log4j.properties

preprod.log4j.properties

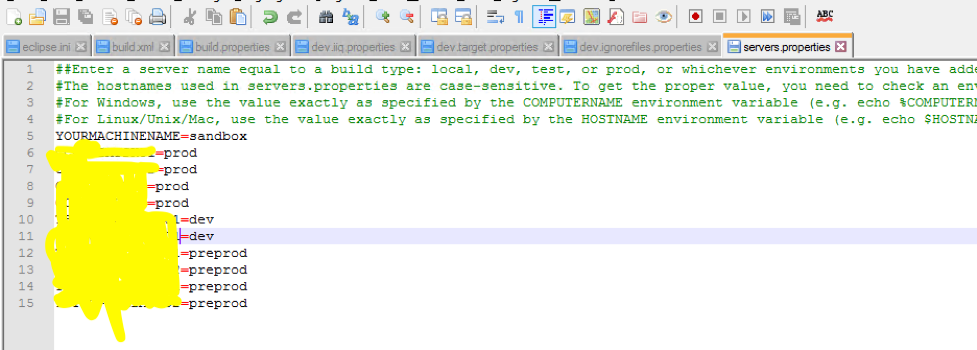
prod.log4j.properties



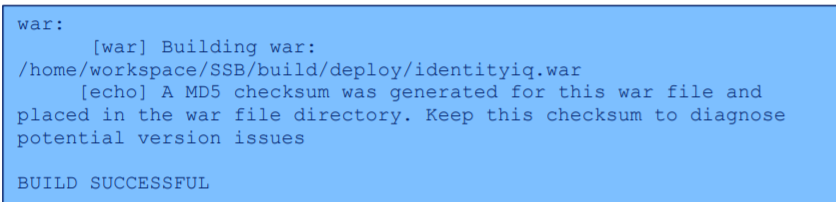
* + - * Attached dev.log4j.properties below

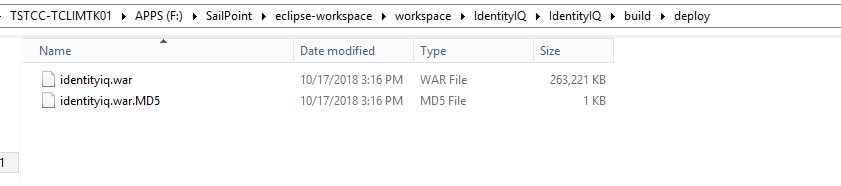


* + - Setting the target variables by editing servers.properties
      * If the SPTARGET environment variable is not set, the build process will attempt to find the environment name by using the servers.properties file, mapping the host where the build is being executed to an environment name. Updated server.properties file as below:



* + - Executing the Build
      * To create a custom IdentityIQ war file (J2EE Web Application Archive) that can be deployed to a web application server such as Tomcat, you can perform the following steps.
        + Open a Terminal or Command Prompt window
        + Navigate to <SSB install directory>\
        + Enter ./**build war**
        + This will generate a deployable war file in your **<SSB install directory>\build\deploy** folder and you will receive a confirmation message like the one below

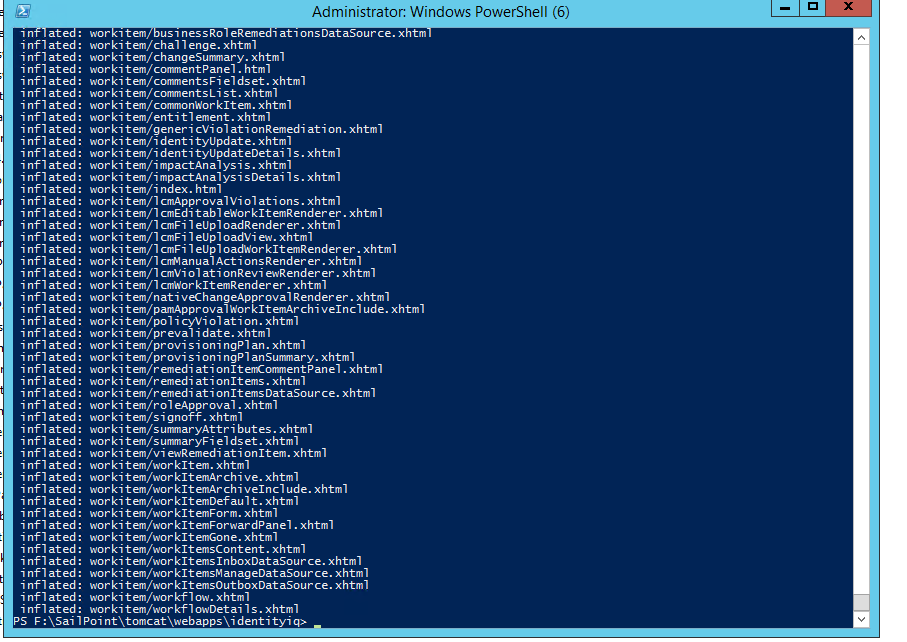




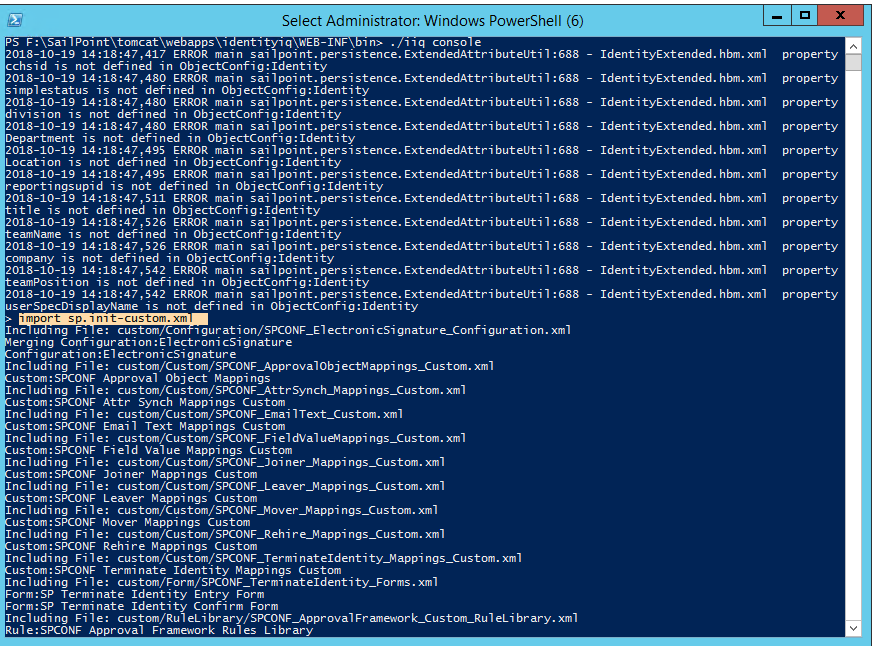
* + - * Deploy this file to the target web application server. You may need to consult your application server’s deployment guide for details. For Tomcat:
        + Copy this custom identityiq.war to a folder under <Tomcat>/webapps

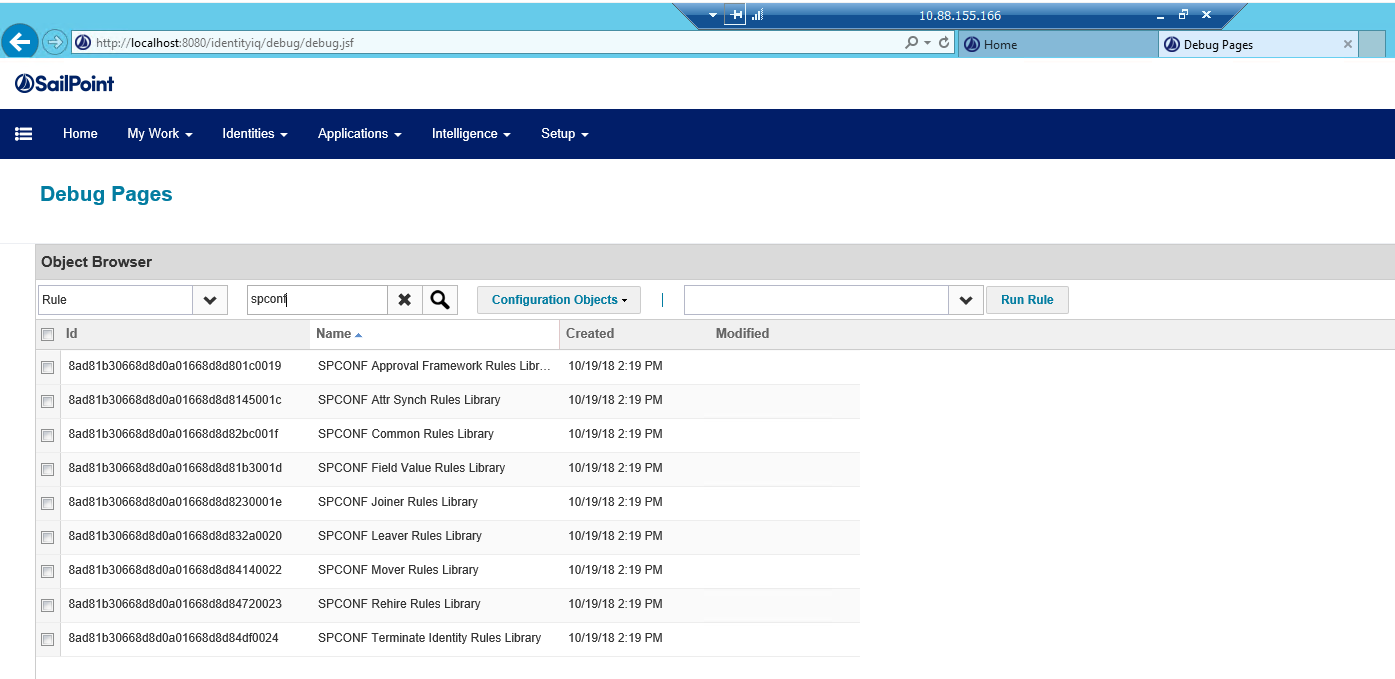
(e.g. <Tomcat>/webapps/identityiq)

* + - * + Navigate to that directory and expand the war: jar xvf identityiq.war



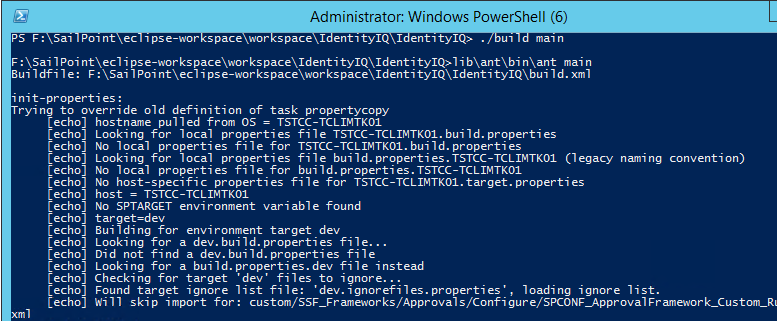
* + - * + Delete the war file once you have expanded it
      * If you wish to update any custom objects and redeploy them to IdentityIQ you can perform the following steps. Open a terminal or command prompt window,
      * Navigate to the <SSB install directory> folder and enter **build importdynamic**. This command will import all the custom XML artifacts from your **config** folder into IdentityIQ. It will utilize the DB connection from <environment>.iiq.properties and import those XML objects into the IdentityIQ database. Target importdynamic will not cycle the application server, a step required if changes are made to any class files included in your SSB directory. An alternative to using build importdynamic is to **manually** import the **sp.init-custom.xml** file that was generated during the build, using the command **import sp.init-custom.xml** inside **iiq console**.



* + - * Open IdentityIQ in a web browser and you will see the applications, rules, and other custom objects from your original environment in this new one. Note that the SSB modifies the init.xml normally used for a “fresh” build of IdentityIQ. The SSB modifies this file from the defaults to import all content (custom objects, LCM objects if desired, and default objects). 

## Setup Deployment Accelerator

* + - 1. Get the SSB project created into eclipse workspace using Git repository
      2. Run build main on the SSB project



* + - 1. Copy the IdentityIQ.jar and commons-logging-1.1.3.jar to lib folder
      2. Update the Class path as shown below.
      3. Update .project file in with below values

<buildCommand>

<name>sailpoint.IIQ\_Deployment\_Accelerator.IdentityIQ.iiqArtifactBuilder</name>

<arguments>

</arguments>

</buildCommand>

<natures>

<nature>sailpoint.IIQ\_Deployment\_Accelerator.IdentityIQ.iiqNature</nature>

<nature>org.eclipse.jdt.core.javanature</nature>

<nature>org.eclipse.wst.jsdt.core.jsNature</nature>

</natures>

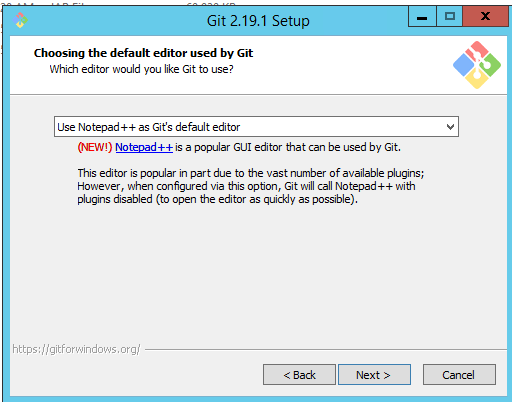
3.

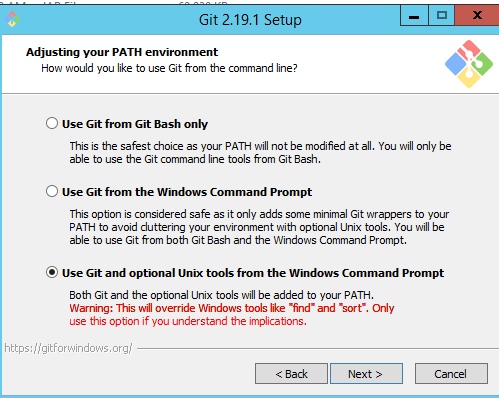
## Install Git and GitLab Runner

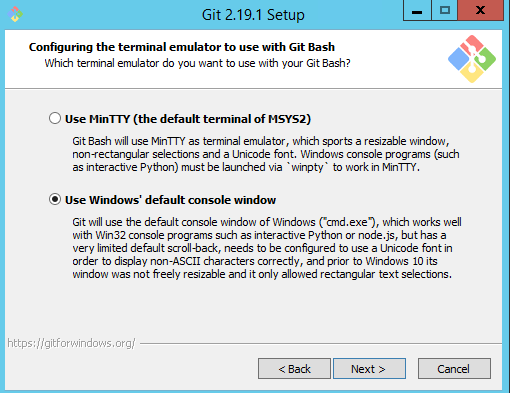
* 1. Download Git from below link for 64bit version

<https://git-scm.com/downloads>

* 1. Run Git as administrator and accept defaults except below changes:







* 1. Click Install

### Command line instructions to Download Repository

##### Open CMD as administrator

##### Execute below Git global setup commands

git config --global user.name "Madiraju, Manjunath"

git config --global user.email [madiram@CLIENT.org](mailto:madiram@ccf.org)

##### Execute below command to NOT validate the certificate

git config --global http.sslVerify false

##### Create a new repository

git clone https://gitlab.CLIENT.org/adgroup/IdentityIQ.git

cd IdentityIQ

touch README.md

git add README.md

git commit -m "add README"

git push -u origin master

##### Checkin base project code to repository

Copy the IdentityIQ base project generated from SSD to the IdentityIQ repository created above.

git add --all

git commit -am "Created base IIQ project with SSB and SSF features"

git push -u origin master

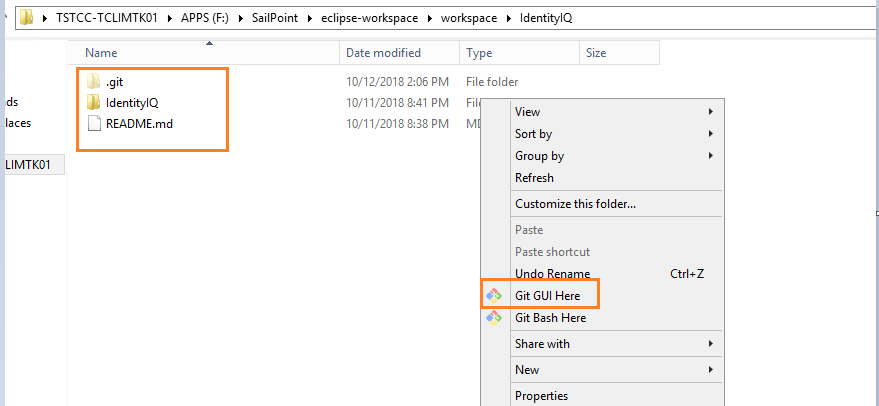
##### Repository is complete

* + 1. If we need to clone a branch repository, repeat steps 1 to 3 and execute below command.

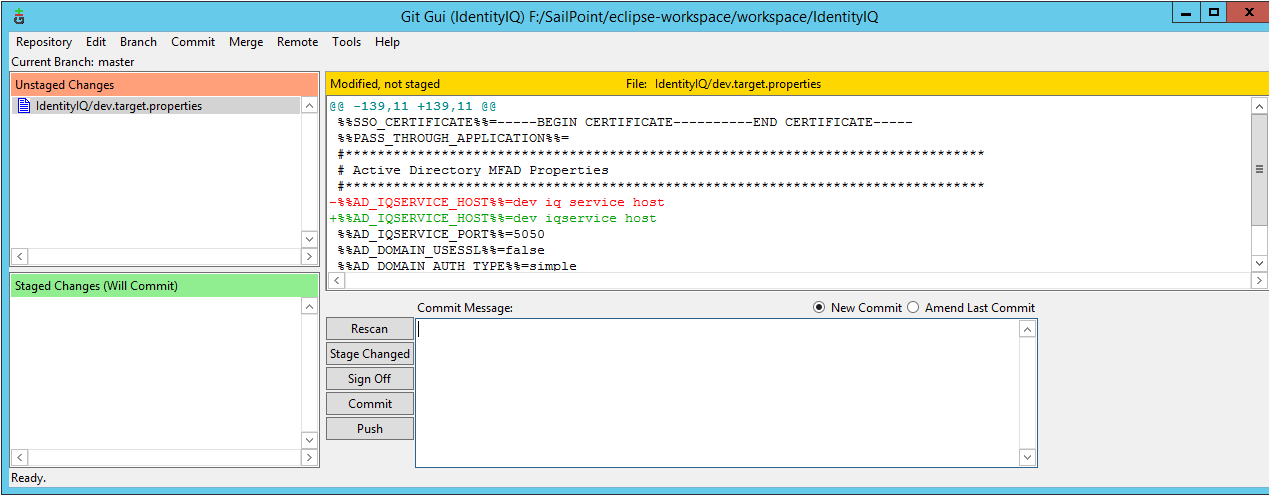
git clone -b develop\_caregiverconnector https://gitlab.CLIENT.org/adgroup/IdentityIQ.git

Install Git GUI

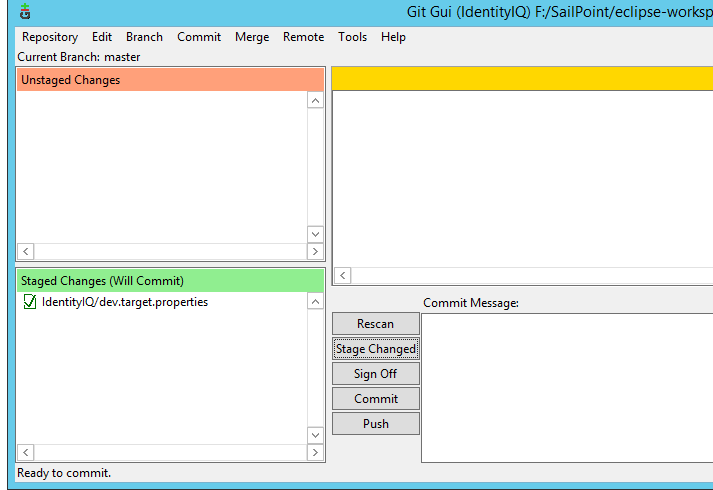
* + - Navigate to the location where you have your project workspace
    - Right click and select Git GUI



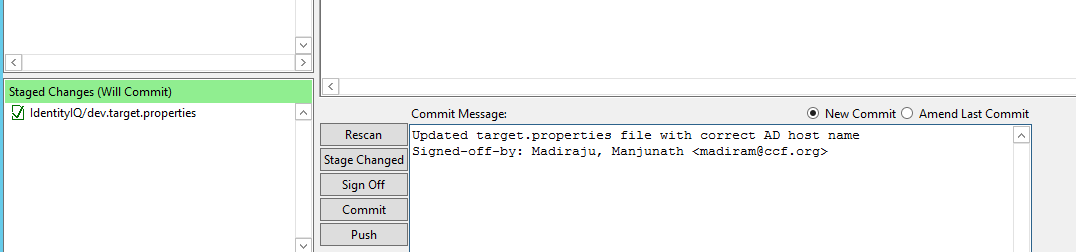
* + - Valid the unstaged changes to make sure it has only your changes. Click Stage Changed button



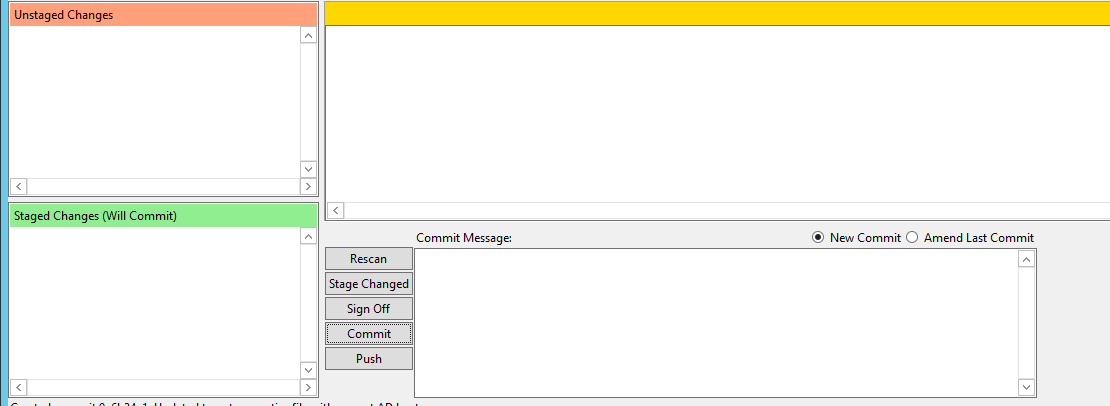
* + - Once you click Stage changed, file will move to bottom section.



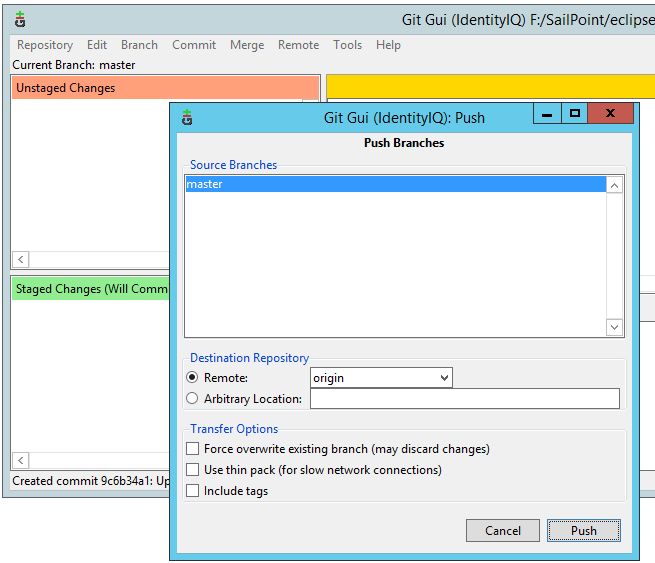
* + - Click Sign Off. Make sure your email id is correct. Enter the commit message.



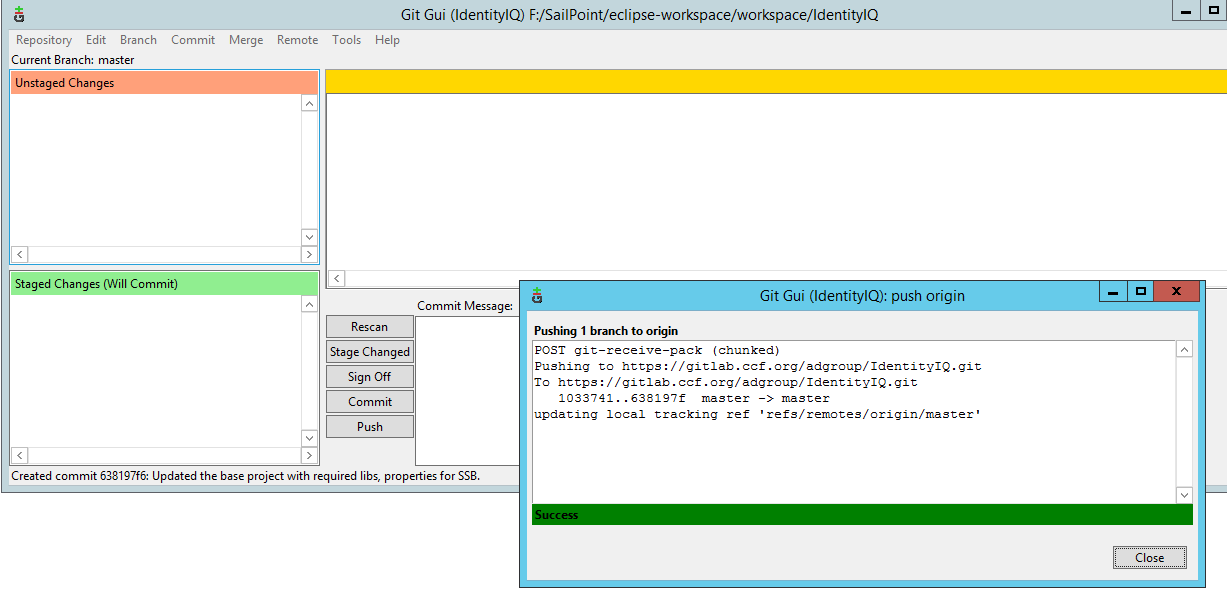
* + - Click Commit. Changes gets committed locally and staged section becomes empty.



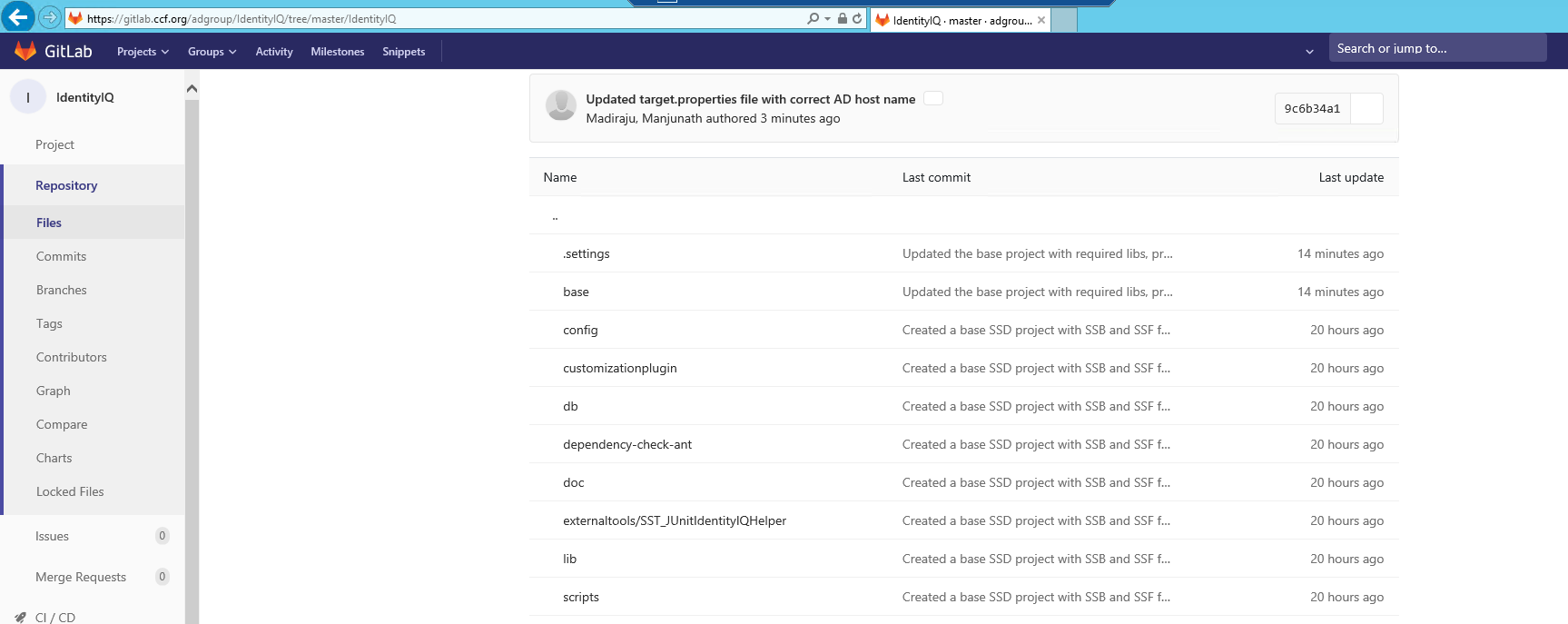
* + - Click Push button and select the branch. In my case its master. But usually it will be the branch on which you are making code changes.



* + - Click Push again.



* + - Login to Gitlab and validate your files are checked in as shown below.



|  |  |
| --- | --- |
| Contact us |  |
| **Ken Dunbar**  Director - Cybersecurity  T 972 342 9772  E kbdunbar@kpmg.com |  |
|  |  |
| ©2018 KPMG LLP, a Deleware limited liability partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative (“KPMG International”), a Swiss entity. All rights reserved.  The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavour to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.  The KPMG name, logo and “cutting through complexity” are registered trademarks or trademarks of KPMG International Cooperative (“KPMG International”). | |