Installing iManager

NetIQ iManager provides a global view of your network from one browser-based tool, allowing you to assess and respond to changing network demands proactively.

Using iManager, you can administer NetIQ eDirectory and many other NetIQ and third-party products, including Novell Open Enterprise Server, NetIQ Identity Manager, Novell Audit, and BorderManager. This centralized management system eliminates administrative redundancy and overhead, saving your time and money.

This section includes the following information:

Server-based and client-based versions of iManager

The traditional server-based version of iManager 2.7.7, called simply iManager, is installed on a server to access an eDirectory tree. The client-based version of iManager, called iManager Workstation, is installed on a client workstation rather than a server. Use the following guidelines to decide which version fits best in your environment or whether your eDirectory management policies would benefit from installing both versions:

iManager Workstation

You can use iManager Workstation in scenarios where you have only one administrator who manages eDirectory from the same client workstation or has multiple administrators and uses customized plug-ins.
You must install plug-ins on each administrator's client workstation as iManager plug-ins do no automatically synchronizes between iManager instances. Some of the features of the iManager Workstation includes:

- iManager Workstation is fully self-contained, and its setup is simple.
- iManager Workstation automatically starts and stops the resources it needs for loading and unloading operations.
- iManager Workstation installs and runs on various Linux or Windows client workstations.
- iManager Workstation has no dependencies on server-based iManager and can coexist with any other versions of iManager installed on your network.

iManager Server

If you manage eDirectory from multiple client workstations or have multiple administrators, install iManager Server so that it is available from any connected workstation. Additionally, customized plug-ins only need to be installed once per iManager Server.

Note: iManager 2.7.7 Patch 11 installs the iManager 277 base with the patch.

Supported products for iManager Server

iManager Server supports the following products:

- Operating systems
- Application servers
- Browsers
- Directory services

Operating systems

The following table contains a list of the certified and supported operating systems that iManager Server can run on.

Note: In the following table, the Certified column indicates that the testing is completed for an operating system and supported.
The Supported column indicates that testing an operating system is pending, but it is expected to work.

Certified Server Operating System Version	**Supported Operating Systems**	**Notes**
Windows Server 2012, Windows Server 2012 R2 (64-bit)	Supported on later versions of service packs	Supported on later versions of service packs
Windows Server 2008 Enterprise Edition SP2 (64-bit)	Supported on later versions of service packs	iManager Server runs only in 64-bit mode.
Windows Server 2008 R2 Enterprise Edition (64-bit)	Supported on later versions of service packs	iManager Server runs only in 64-bit mode
Windows Server 2008 R2 Standard Edition (64-bit)	Supported Edition (64-bit) Supported on later versions of service packs	iManager Server runs only in 64-bit mode

Application Servers

iManager Server supports Tomcat 7.0.81

Note: Do not install standalone iManager package on a Tomcat server running other applications.

Browsers

iManager Server supports the following browsers:

- Apple Safari 8.0 and 7.0.6 Google Chrome 38, 37, 31, 30, 28, 27, 26, 25, 23, and 22
- Internet Explorer 11, 10 (Normal and Compatibility modes), 9, and 8
- Mozilla Firefox 33, 32, 26, 25, 23, 22, 21, 19, 16, 15, 13, 12, 11, 10, 9.0.1, and 4.0.1

Directory Services

iManager supports eDirectory 8.8 and later Support Packs.

Self-Signed Certificates

Standalone iManager installations include a temporary, self-signed certificate for use by Tomcat. It has an expiration date of one year. This approach is not intended to be a long-term implementation. It is a temporary solution to get your system up and running so you can securely use iManager immediately following installation. OpenSSL does not recommend using self-signed certificates except for testing purposes.

One challenge to replacing the self-signed certificate is Tomcat's default keystore uses in Tomcat (JKS) format. The tool used to modify this keystore, keytool, cannot import a private key. It will only use a self-generated key.

If you are using eDirectory, you can use Novell Certificate Server to securely generate, track, store and revoke certificates with no further investment. To generate a public/private key pair in eDirectory using Novell Certificate Server, complete the following steps for your applicable platform:

Note: This section's information does not apply to OES Linux, which installs both Tomcat and Apache. The OES Linux documentation includes details on replacing the self-signed Apache/ Tomcat certificate. See, Setting Up Certificate Management in the OES 11 SP2: Planning and Implementation Guide.

Linux

The following instructions show how to create a keypair in eDirectory and export the Public, Private, and Root Certificate Authority (CA) keys via a `PKCS#12` file on the Linux platform. It includes modifying Tomcat's `server.xml` configuration file to use the PKCS12 directive and point the configuration to an actual P12 file rather than use the default JKS keystore. This process uses the following files:

- The temporary keypair is held in the `/var/opt/novell/novlwww/.keystore` file.
- The trusted roots are contained in the `/opt/novell/jdk1.7.0_25/jre/lib/security/ cacerts` file.
- The file for configuring Tomcat's use of certificates is `/etc/opt/novell/tomcat7/server.xml`.

Procedure

1. Create a new server certificate with iManager.
   In iManager, select **Novell Certificate Server** > **Create Server Certificate**.
   Select the appropriate server, specify a nickname, and accept the rest of the certificate defaults.
2. Export the server certificate to the Tomcat home directory (`/var/opt/novell/novlwww`)
3. Convert the `.pfx` file to a `.pem` file.

   To do this, use a command similar to the following:
   ```
   openssl pkcs12 -in newtomcert.pfx -out newtomcert.pem
   ```

   Specify the certificate password specified in Step 2, and then enter a password for the new .pem file. You can use the same password if required.
4. Convert the .pem file to a .p12 file.
   To do this, use a command similar to the following:
   ```
   openssl pkcs12 -export -in newtomcert.pem -out newtomcert.p12
   -name "New Tomcat"
   ```
   Specify the certificate password specified in Step 3, and specify a password for the new `.p12` file. You can use the same password if required.
5. Enter the following command to stop Tomcat:
   ```
   /etc/init.d/novell-tomcat7 stop
   ```
6. Edit the Tomcat configuration file `(/etc/opt/novell/tomcat7.0.42/server.xml)` and add `keystoreType`, `keystoreFile`, and `keystorePass` variables to let Tomcat use the newly created `.p12` certificate file. For example:
   ```
   <Connector
   className="org.apache.coyote.tomcat7.CoyoteConnector"
    port="8443" minProcessors="5" maxProcessors="75"
   enableLookups="true"
    acceptCount="100" debug="0" scheme="https" secure="true"
    useURIValidationHack="false" disableUploadTimeout="true">
    <Factory
   className="org.apache.coyote.tomcat7.CoyoteServerSocketFactor
   y"
   clientAuth="false" protocol="TLS" keystoreType="PKCS12"
   keystoreFile="/var/opt/novell/novlwww/newtomcert.p12"
   keystorePass="password" />
   ```

```
</Connector>
```

7. Change the `.p12` file's ownership to the appropriate Tomcat user/group, typically `novlwww`, and set the file permissions to `user=rw`, `group=rw`, and `others=r`. For example:
   ```
   chown novlwww:novlwww newtomcert.p12
   chmod 654 newtomcert.p12
   ```
8. Enter the following command to restart Tomcat:
   ```
   /etc/init.d/novell-tomcat7 start
   ```