Infor ION Grid Security Administration Guide

Version 11.1.x
Published February 19, 2014

Publication Information

Release: 11.1.x

Publication date: February 19, 2014

Document Number: GRIDSECAG_11.1.x_UWA_09

Contents

Introduction

1

About This Guide

Purpose

The aim of this guide is to explain the security-related concepts and procedures for the grid that an administrator needs to know.

Knowledge Prerequisites

The reader of this document is expected to have some basic administrative knowledge of the grid and grid applications. The reader is also expected to have basic knowledge regarding certificates, public key cryptography (asymmetric encryption), and certificate authorities.

For administrators of grids that are installed and managed through LifeCycle Manager, it is expected that the administrator is familiar with LifeCycle Manager.

Which Grids Does This Guide Apply To?

This administration guide applies to two grids: the ION Grid (current version 11.1.10.0) that is installed via a Java jar installation program and the ION Grid (current version 11.1.11.0) that is uploaded and installed through LifeCycle Manager. The ION Grid that is installed by the Java program does not use the LifeCycle Manager. Thus, any procedures that refer to the LifeCycle Manager only apply to the ION Grid that is installed through the LifeCycle Manager. Procedures that refer to the Grid Management Page, the Grid Configuration Manager, or command line tools apply to both grids.

Throughout this guide, the term "ION Grid for LifeCycle Manager" will refer to the ION Grid that is uploaded, installed, and managed through LifeCycle Manager.

The term "ION Grid" will refer to both ION Grids. This is because procedures that apply to the ION Grid that is installed via the Java program also apply for the most part to the ION Grid for LifeCycle Manager. If a concept or procedure applies only to the ION Grid that is installed via the Java program, that limitation will be made explicit.

For non-security administration documentation for the ION Grid, see the *Infor ION Grid Administration Guide*, for version 11.1.10.0 of the ION grid (installed through the Java program) and the *Infor ION Grid Administration Guide for LifeCycle Manager 10*, for version 11.1.11.0 of the ION Grid.

The *Infor ION Grid Security Administration Guide* does not apply to the Lawson Grid. For administration and security documentation for the Lawson Grid, see the *Lawson Grid Administration Guide* and the *Lawson Grid Extensions Installation and Administration Guide*.

ION Grid Security Overview

Most grid security considerations are related to communication – communication within a grid, as well as to and from different types of grid clients. To maintain the security of this communication, the grid uses certificates for communication within a grid. In addition, for communication involving grid clients, it uses both certificates and other types of authentication, such as those provided through session providers. You can further use session-based authentication and role restrictions to control access to methods or functionality within a grid.

Certificates in the Grid

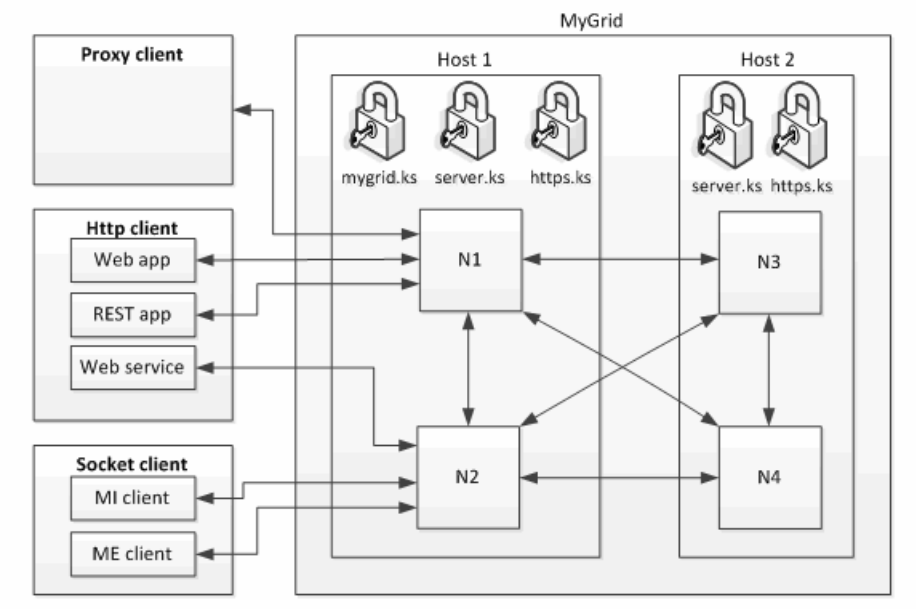
The figure below shows the different paths of communication within and to an example grid. The grid is called MyGrid, and it has two hosts and four different nodes. Keystores are denoted by padlock symbols.

Each grid has a master keystore called [gridname].ks. In this illustration, the grid is an ION Grid installed through the Java installation program, so this keystore (mygrid.ks) resides on Host 1. Mygrid.ks contains the grid root key pair, which is used to sign certificates in this grid.

Additionally, both hosts have their own server.ks keystore, which contains a host-specific key pair and a certificate signed with the grid root key pair. These credentials are used for communication between the nodes in the grid (N1-4), as well as for securing proxy calls.

Finally, there are host-specific https.ks keystores on both hosts. These contain the credentials used for SSL server authentication, when clients connect to the grid.

Figure 1. Illustration: Sample grid with keystores



Grid-provided Services

As shown in the figure in "Certificates in the Grid" on page 7, a grid can be accessed by different types of clients. These include proxy clients, HTTP clients (web application, REST application, and web service), and socket clients (MI and ME clients). These clients access different types of grid-provided services.

The client connections are secured in the following manners:

- Proxy client

  Grid proxy calls are transmitted over the grid proxy protocol, which in turn uses sockets or secure sockets. The server authenticates with the key material in server.ks. Clients may authenticate their users with a certificate signed by the grid root key, or through creating a session by calling a session provider (see "Grid Principals and Sessions" on page 25) once the connection has been established.

- Web and REST applications

  Web and REST applications can be accessed via HTTP or HTTPS, in which case the connection can be either server-authenticated (using the key material in https.ks) or mutually authenticated. Clients may authenticate their users with a certificate signed by the grid root key. It is also possible to authenticate using regular HTTP transport security, for example, basic, digest, Kerberos, and so on.

- Web service

  Web services use the same security mechanisms as web and REST applications. Additionally, Web Services Security with username and token may also be used.

- Socket clients

  Socket clients such as MI and ME (both of which are used in an M3 context) may use SSL, or an application-specific protocol to establish a session.

Note: In order for server authentication to succeed, the client must trust the grid root key.

Authentication

In addition to the certificate-based authentication mentioned in "Certificates in the Grid" on page 7, the grid provides a pluggable architecture for using different sources and protocols for authentication such as LDAP, ADFS 2.0, NTLM, or Kerberos. These services are provided by grid session providers. There are several different types of session provider, depending on the scenario.

When a user has been authenticated by a session provider, a grid session is created for that user.

For more information, see "Authentication Overview" on page 24 and "Grid Principals and Sessions" on page 25.

Authorization

Methods in the grid may be protected in one of several ways. The authorization level that mostly affects users is the session-based authorization with or without role restrictions. Plain session-based authorization means that a user must have a valid grid session (that is, the user must be authenticated). When role restrictions are added, the authenticated user needs to have at least one of a predetermined set of roles. These roles are mapped in the Configuration Manager, using information obtained from the authentication source.

For more information, see "Authorization Overview" on page 60.

ION Grid Certificate Management

2

Grid Keystores

The certificates used in the grid are partitioned into several single-usage keystores. It is important to protect the keystore files and the keystore password files by applying proper file permissions.

In an ION Grid installed through the Java installation program, this protection is set up automatically at installation time.

Some grid applications may have their own keystores. Please review the documentation for those applications when necessary.

gridname.ks

The keystore *gridname*.ks contains the grid root key pair and certificate. In a LifeCycle Manager-controlled grid, this keystore is stored on the LifeCycle Manager server to ensure security.

In an ION Grid installed through the Java installation program, this keystore is only available on the first host installed.

The private key in this keystore is used for signing host, client, and SSL certificates used in the grid, as described in the following sections.

It is recommended to back up this keystore and the password file that belongs to it.

server.ks

The server.ks keystore, found on each host, contains a certificate and the corresponding private key used for internal grid communication. Each host in a grid has its own server.ks. The host certificate in server.ks is signed by the grid root key to enable inter-grid host trust. The server.ks keystore is sometimes referred to as the host keystore.

http.ks

The https.ks keystore, found on each host, contains the SSL certificates used on that host. Each host must have its own https.ks keystore. By default, the certificate in the keystore is signed by the grid root key, but it is possible to create a Certificate Signing Request (CSR), have an external Certificate Authority sign the request, and import the signed certificate into the keystore. For instructions, see "Creating Certificate Signing Requests and Importing Certificates" on page 18.

HTTPS/SSL Certificates

When a grid client connects to the grid using HTTPS, the default behavior for an ION Grid for LifeCycle Manager is server authentication only. The server (that is, the grid host) authenticates to the client using the SSL certificate in https.ks. In order for this authentication to work, the client must trust the signer of this certificate, either the grid root certificate or an external CA.

In an ION Grid installed through the Java installation program, the default setting is that clients may authenticate with a certificate.

The SSL certificates in https.ks are used by the routers in the grid (the routers are the HTTPS endpoints). It is not necessary (or even possible) to have separate certificates per web service. The grid web services are not aware of the certificates in the routers.

It is recommended to have the client browser either trust the grid root certificate, or use SSL certificates signed by a CA already trusted by the client browsers.

Securing Grid Proxy Connections

The grid proxy protocol used by clients for programmatic access to the grid can be configured to run over SSL (TLSv1). The grid server authenticates with the key material in server.ks. The connection allows client authentication but does not require it. If client authentication is desired (and the client

application supports it) use a grid client certificate, generated with the console method **`-create=clientcert`** (see "Console Tool Guide" on page 12).

To configure SSL for grid proxy clients

1. Access the Configuration Manager and click Routers.
2. Select the router you wish to configure.
3. Check the Encryption option. SSL will now be in use for grid proxy connections to the port indicated by the Port setting.
4. To enable specific cipher suites for the SSL connection, use the Optional Cipher Suites text box. Use the format of the JSSE Cipher Suite Names from the Java Cryptograhy Architecture Standard Algorithm Name Documentation, for example, **`TLS_RSA_WITH_AES_128_CBC_SHA`**. If any cipher suites are specified, the server will offer those cipher suites (only) when negotiating the protocol during the SSL handshake. If the box is left blank, the client and server base the protocol negotiation on the available cipher suites in the JDKs of the client and server, respectively.

Certificate Authority Functionality

The grid contains a basic Certificate Authority used to generate and sign certificates for the use of that grid only. Some of this functionality applies to grids installed through the Java installation program and some of it applies both to LifeCycle Manager installations of the grid and to grids installed through the Java installation program.

The grid root CA certificate is generated during installation. It is only used to sign other certificates. The grid CA can also issue Certificate Signing Requests that can be submitted to external Certificate Authorities to create a certificate signed by that CA. The only use for that today is for making SSL certificates, which can be automatically trusted by users' browsers.

Console Tool Guide

The grid provides console tools to generate CSRs for an external CA to sign, to create grid-signed certificates, and to import certificates. These certificates are then used in a router for SSL connections to a specific host. To create CSRs and certificates or to import certificates using the console tool, see the "Console Tool Methods and Options" section below.

Console Tool Methods and Options

There are console methods available for manually creating certificate signing requests for external signing, as well as methods for creating grid-signed certificates.

The console methods can be called with the following command:

UNIX, Windows, and IBM i:

java -cp bcmail-jdk*BCVersion*.jar;bcprov-jdk*BCVersion*.jar;grid-core-*GridVersion*.jar com.lawson.grid.security.Certificates -create...

Linux:

java -cp bcmail-jdk*BCVersion*.jar:bcprov-jdk*BCVersion*.jar:grid-core-*GridVersion*.jar com.lawson.grid.security.Certificates -create...

where *BC version* is the version of your Bouncy Castle and *GridVersion* is the version of the grid you installed. To get the exact names of these three jar files, look in the /runtimes/*gridVersion*/resources folder in the grid installation folder.

There are two methods and a number of options available for managing the certificates.

Console Methods

Two methods are available in the console mode: create and import. The create command generates certificates or certificate requests. The import command imports a certificate after it has been signed by an external CA.

The create command

UNIX, Windows, and IBM i:

java -cp bcmail-jdk*BCVersion*.jar;bcprov-jdk*BCVersion*.jar;grid-core-*GridVersion*.jar com.lawson.grid.security.Certificates -create[=<gridcert|hostcert|clientcert|sslcert|certreq|symkey>]

Linux:

java -cp bcmail-jdk*BCVersion*.jar:bcprov-jdk*BCVersion*.jar:grid-core-*GridVersion*.jar com.lawson.grid.security.Certificates -create[=<gridcert|hostcert|clientcert|sslcert|certreq|symkey>]

This command is used to generate certificates or certificate requests. The create method requires a command option which indicates what to create.

Command Option	Description
gridcert	The **create=gridcert** command creates a new self-signed grid root certificate/keypair and stores them in the provided grid keystore. If a grid root certificate is generated after the initial installation, it is necessary to regenerate all certificates that were signed by the previous grid root certificate. This includes all host, client, and SSL certificates that were not signed by an external CA. Note that the corresponding grid root password is not stored automatically when using this command. If you wish to create a new password, this must be stored in a file called *gridName*.pw next to the grid root keystore. It is also possible to reuse the existing *gridName*.pw by entering its contents as the -gridpassword argument. **Note:** This command should only be used when absolutely necessary. For more information, see "Renewing/Reissuing Grid Certificates" on page 22.
hostcert	The **create=hostcert** command creates a new grid-signed host certificate for a grid host to be permitted to participate in a grid, and to communicate with the other hosts in that grid. Host certificates require the grid-admin role. Note that the symmetric key must also be regenerated when a new host certificate is created.
clientcert	The **create=clientcert** command creates a new grid-signed client certificate used to authenticate to the grid. This can be used in the scenarios described in "Grid-provided Services" on page 8. The roles provided when creating the certificate influences the permissions the user will have when connecting with that certificate. Note that the corresponding client keystore password is not stored automatically when using this command. If you wish to create a new password, this must be stored in a file called *clientName*.pw next to the client keystore. It is also possible to reuse the existing *clientName*.pw by entering its contents as the -clientpassword argument.
sslcert	The **create=sslcert** command creates a new grid-signed SSL certificate to be used by the routers of a specific grid host.

Command Option	Description
certreq	The **create=certreq** command creates a Certificate Signing Request (CSR) for an SSL server certificate. This CSR is then sent to an external Certificate Authority for signing. When the certificate has been signed, it can be imported using the **import=sslcert** command.
symkey	The **create=symkey** command is used to generate the secret key a grid host needs in order to decrypt/encrypt grid password properties. Note that existing password properties will not decrypt correctly if the symmetric key is regenerated. To avoid this, export the properties via the Configuration Manager before regenerating the symmetric key, and then import them back afterwards.

The import command

UNIX, Windows, and IBM i:

java -cp bcmail-jdk*BCVersion*.jar;bcprov-jdk*BCVersion*.jar;grid.jar com.lawson.grid.security.Certificates -import[=<sslcert>]

Linux:

java -cp bcmail-jdk*BCVersion*.jar:bcprov-jdk*BCVersion*.jar:grid.jar com.lawson.grid.security.Certificates -import[=<sslcert>]

This command is used to import an SSL certificate after it has been signed by an external CA. The same keystore must be specified during the import as when the Certificate Signing Request was generated.

Console Method Options

Most of the following options are used in conjunction with the **create** command. A <p> after the option name indicates that a value is needed for the option.

Method Option	Description
-address <p>	An IP address that this certificate is valid for. Host names will be resolved using java.net.InetAddress.getByName(). May be specified more than once to enable the certificate to be valid for multiple IP addresses.
-altname <p>	Alternate fully qualified domain name or the IP address for the host for which this certificate is created. May be specified more than once to enable certificate to be valid for multiple fully qualified domain names.

Method Option	Description
-certfile <p>	The name of a file containing a signed certificate and/or certificates needed to establish a trust chain to the signed certificate. May be specified more than once.
-clientkeystore <p>	Path to the directory where keystore file for the client is to be saved.
-clientname <p>	Name of the client for which this certificate is created.
-clientpassword <p>	Client certificate password.
-dname <p>	The X.500 distinguished name to be used in the subject field in the certificate. It should not contain the CN attribute since this is automatically added, derived from the hostfqdn option.
-from <p>	Certificate valid from date in YYYYMMDD format (default today).
-gridkeystore <p>	Path to the grid keystore directory.
-gridname <p>	Name of the grid for which this certificate is created.
-gridpassword <p>	Grid certificate password.
-hostfqdn <p>	The fully qualified domain name or the IP address for the host for which this certificate is created.
-hostkeystore <p>	Path to the host keystore directory.
-hostname <p>	Name of the host for which this certificate is created.
-keyalg <p>	Specifies the algorithm to be used to generate the keypair. The default is **RSA**.
-keysize <p>	Specifies the size of each key to be generated. The default is **1024**.
-keystoretype[=<bks|jks|pkcs12>]	Keystore type for client keystores, optional; default is **jks**.
-role <p>	A role attached to this principal. May be specified more than once. Only valid for client certificates.
-serial <p>	Certificate serial number, uniquely maintained by the CA.
-sigalg <p>	Specifies the algorithm that should be used to sign certificates and CSRs. The default is **SHA256WITHRSA**.
-sslkeystore <p>	Path to the SSL keystore directory.
-symkeypath <p>	Path to the host secret key directory.
-to <p>	Certificate valid to date in YYYYMMDD format (default in 90 days in the future).

Method Option	Description
-unresolved	Modify the meaning of address to not resolve host names.

Example Console Commands

Command to create a new grid root certificate for the Grid called "demoGrid". The key size is set to 2048 bits using the default key algorithm and signature algorithm. The key will be valid from today until the 20th of November 2022. The keystore will be saved in the "ks folder":

```
-create=gridcert –gridkeystore <ks folder> -gridname demoGrid
-gridpassword <password> -keysize 2048 –serial <unusedSerialNumber> –to 20221120
```

Command to create a host certificate for the server demogrid.infor.com. The keystore is saved in the "ks folder". Default key size and algorithms are used.

```
-create=hostcert -hostfqdn demogrid.infor.com -to YYYYMMDD –hostkeystore
<ks folder> -hostname demogrid –gridkeystore <ks folder> -gridpassword
<password> -gridname demoGrid  -serial <unusedSerialNumber> -address <IPAddress> -role grid-
admin
```

Command to create an SSL certificate for a server with multiple network interfaces (demogrid.infor.com using IP address 10.10.10.10 and extdemo.infor.com using IP address 172.30.10.10). The keystore is saved in the "ks folder". Default key size and algorithms are used.

```
-create=sslcert -address 10.10.10.10 -address 172.30.10.10 -altname
demogrid.infor.com -altname extdemo.infor.com -to YYYYMMDD -sslkeystore
<ks folder> –gridkeystore <ks folder> -gridpassword <password>
-gridname demoGrid -hostfqdn <hostFQDN>
```

Command to create a client certificate for the user "MyID" with the roles "grid-admin" and "other-role". If the YYYYMMDD values were "20130601" and "20150615", the certificate would be valid from June 1st 2013 to June 1st 2015.

```
-create=clientcert –clientkeystore <ks folder> -clientname MyID
-clientpassword <password> -from <YYYYMMDD> -to <YYYYMMDD> -role grid-admin
-role other-role –gridkeystore <ks folder> -gridpassword <password>
-gridname demoGrid –serial <unusedSerialNumber>
```

Command to create an SSL Certificate Signing Request. The CSR is written to the file <ks folder>/<hostname parameter>.csr.txt. The CSR can then be used in the external CA to get a signed certificate back.

```
-create=certreq -address 10.10.10.10 -address 172.30.10.10 -altname
demogrid.infor.com -altname extdemo.infor.com –hostname demogrid -sslkeystore <ks folder>
-hostfqdn <hostFQDN> –serial <unusedSerialNumber>
```

Command to import an externally signed certificate. The <ks folder> must be the same path as was used during the create=certreq command.

```
-import=sslcert -certfile <path to a file containing the signed certificate> -sslkeystore <ks
folder>
```

Command to create a new symmetric key:

```
-create=symkey -gridpassword <password> -gridname <gridName> -gridkeystore <ks folder>
-symkeypath <ks folder> -hostkeystore <ks folder> -hostname <hostName>
```

Grid-signed vs. CA-signed Certificates

The default behavior of the grid is that grid server and client certificates are signed with the grid root key. In the case of grid SSL server certificates, it is also possible to have them signed by an external CA.

The benefit of having a CA-signed certificate is that clients automatically trust the issuer of the certificate if the CA is one that the clients already trust. This is the case for public Certificate Authorities (such as VeriSign, EnTrust, Thawte, and so on).

In many organizations, it is easy to get the grid root certificate trusted by the browsers of their own organization. It might be trickier when accessing using different handheld devices or for uncontrolled devices (for example, external users).

The decision to use grid-signed or CA-signed certificates depend on the use of grid and the applications that run in the grid.

Note that a multi-host grid may have CA-signed SSL certificates on some hosts and grid-signed SSL certificates on others, depending on what each host/router is used for.

When to Use Grid-signed SSL Certificates

Grid-signed certificates are suitable in certain scenarios, for example, test installations or installations that only have managed clients where it is easy to ensure that all clients automatically trust the grid root certificate.

When to Use CA-signed SSL Certificates

It is recommended to use CA-signed SSL certificates in any scenario where it is impossible or simply too much work to get connecting clients to trust the issuing certificate. Scenarios can be:

- When using a grid installation running in the cloud.
- Internet-facing routers (not recommended), or routers that external users can connect to using unmanaged devices (for example using VPN connections).

Creating Certificate Signing Requests and Importing Certificates

To be able to have SSL certificates in the router that are automatically trusted, an external CA can be used. Certificate signing requests to be signed by an external CA can be created in two ways: using console tools or the ION Grid Configuration Manager.

To create a certificate signing request via the console tools

1 Generate a certificate signing request on the grid host where the externally signed SSL certificate should reside. Each grid host requires its own SSL certificate if there are any grid routers on that host. For instructions on how to generate the certificate signing request, see "Console Tool Methods and Options" on page 12.

2 Copy the certificate signing request generated in step 1 to the Certificate Authority. See the documentation for the CA on how to submit a certificate signing request.

3 Save the result to file and copy the file to the grid host where the certificate signing request was initially generated.

4 Import the signed SSL certificate. For instructions on how to import it, see "Console Tool Methods and Options" on page 12.

To create a certificate signing request via the Configuration Manager

1 Access the Configuration Manager for the grid.

2 Click Advanced Configuration and then Certificates.

3 For the desired host, create a certificate signing request:

a Click "Manage Certificates" for the desired host.

b Click Create Certificate Signing Request (CSR).

c On the Certificate Signing Request form, enter the following values:

Host FQDN (CN)	Make sure Host FQDN matches the grid host name.
Organization Unit (OU), Organization (O), Locality Name (L), State or Province (ST), Country (C)	Make sure these fields are filled in as expected by your certificate signing service.
Alternative Names	If you plan to use aliases or load balancers, add these names as alternative names.

d Click "Create Request Reuse Keys" or "Create Request Overwrite Keys". The first choice keeps the previous SSL key pair. The existing certificate remains in place until the CSR has been signed and the resulting certificate imported. The second choice immediately overwrites the existing key pair and replaces it with a temporary grid-signed certificate until the CSR has been signed and the resulting certificate imported.

e Select Yes if given a warning that a temporary certificate is generated.

f Download or copy the binary code and click OK on the screen to close the Create Certificate Signing Request (CSR) screen. Note that the expiration date of the temporary generated certificate, if applicable, is 90 days in the future.

g Use the generated binary code to request a CA-signed certificate at your certificate service.

Note: The grid will also require the root certificate, so make sure to get the complete certificate chain or have the root certificate available separately.

4 For the host, import the certificate that was generated for that host. For information on importing, see "Importing Signed SSL Certificates via the Configuration Manager" on page 20.

Importing Signed SSL Certificates via the Configuration Manager

When a CSR has been signed by a Certificate Authority, the resulting certificate chain must be imported to the host where the CSR was generated. Imported certificates must be DER or Base64 encoded.

1 In the Configuration Manager, click Advanced Configuration and then Certificates.

2 Click Manage Certificates for the host that the CSR was generated for.

3 Click Import Signed SSL Certificate.

4 Browse to the file containing the SSL certificate chain and click Import Certificate.

5 After verifying the contents on the Import Signed Certificate window, click Import.

Importing Trusted Certificates via the Configuration Manager

The certificate of an external CA can be imported into the grid host SSL truststores to ensure that the grid hosts trust that Certificate Authority. Imported CA certificates must be DER or Base64 encoded.

To import trusted certificates via the Configuration Manager

1 In the Configuration Manager, click Advanced Configuration and then Certificates.

2 For each host you wish to add the certificate to, click Manage Certificates and perform the following steps.

3 Click Import Trusted Certificate.

4 Browse for the certificate.

5 Click View certificate.

6 Click Import.

Creating an SSL Client Keystore in ION Grid for LifeCycle Manager

You can use LifeCycle manager to generate a client keystore for SSL authentication. This keystore will contain a key pair for the client and a certificate signed by the grid root private key. If the keystore type is JKS or BKS, it will also contain the certificate for the grid root key pair.

Note: In an ION Grid installed through the Java installation program, the same result can be achieved by the use of the console method: **`-create=clientcert`**. For more information, see "Console Tool Guide" on page 12.

To create an SSL client keystore in ION Grid

1 In LifeCycle Manager, locate the grid you wish to connect to.

2 In the Applications view, right-click on the grid and select Grid Maintenance > Manage Security.

The Manage Security dialog box is displayed.

3 Select the Generate client keystore radio button and click Next.

4 In the Create Client Keystore window, consider the following fields:

Keystore name	This is the name of the user to be authenticated via SSL. It is also the name of the keystore itself.
Keystore password	Select a strong password and make sure to remember it since it cannot be retrieved later.
Role list separated by	This should be a list of all roles the user should be assigned when using the keystore for authentication. If no specific role is required, simply enter the username.
Keystore type	The format in which the keystore will be exported. You can select one of the following values: • JKS - Java Keystore, the native keystore type for Java applications. File extension (.ks) • BKS - Bouncy Castle Keystore, provided by the Bouncy Castle crypto provider. Works with Java and especially with applications developed for Android devices. File extension (.bks) • PKCS12 - A standard format, published by RSA Laboratories and usable in a Windows environment. File extension (.p12)
Directory to store the keystore in	This is a directory on the machine that the LifeCycle Manager client is running on.

5 Click Next and then Finish.

The keystore with the generated credentials is written to the selected directory.

Exporting the Grid Root Certificate in ION Grid for LifeCycle Manager

In LifeCycle Manager, you can export the self-signed certificate for the grid root key pair. This certificate can be installed in clients to enable trust in the grid, for example, when using server-authenticated SSL.

1 In LifeCycle Manager, locate the grid you wish to connect to.

2 In the Applications view, right-click on the grid and select Grid Maintenance > Manage Security.

The Manage Security dialog box is displayed.

3 Select the Export Grid SSL certificate radio button and click Next.

4 In the Export properties window, select the folder to store the certificate. This is a directory on the machine that the LifeCycle Manager client is running on.

5 Click Next and then Finish.

The grid root certificate is written to the selected directory.

Renewing/Reissuing Grid Certificates

There are certain scenarios where the integrity of the entire grid must be reinitialized. For example:

- The grid root certificate has expired or is close to expiration.
- The grid root certificate or the keystore password file have been lost or damaged beyond recovery.

If either of these is the case, the grid root certificate and all certificates issued by it must be regenerated.

In order to reinitialize the grid integrity, the steps outlined below need to be performed. For details on getting a certificate and importing it for a LifeCycle Manager installed grid, see the "To create a certificate signing request via the Configuration Manager" procedure in "Console Tool Guide" on page 12. For grids installed through the Java installation program, the steps for getting a certificate and importing it must be performed manually, using the console commands described in "Console Tool Methods and Options" on page 12.

The host and SSL keystores should be placed in the /secure folder of the grid on each host. The grid root keystore should also be placed in this folder, but only on the registry host.

Caution: This procedure changes all keys and certificates used by the grid and should only be performed as a last resort.

To reinitialize grid integrity

1. Using the Configuration Manager, export the application settings of all applications with password properties. Unless you do so, password properties will not be able to decrypt correctly after reinitialization.
2. Stop the grid and all grid hosts.
3. Create a new grid root certificate.
4. For each grid host, reissue the host server certificates for inter-grid communication.
5. For each grid host, reissue the symmetric key used for encryption of password properties.
6. For each grid host, reissue the client certificate of the bootstrap service. To do this, first generate a client certificate with client name "Bootstrap," and then rename the generated keystore to "client.ks". These files should be placed in the installation /secure folder.
7. Reissue all client certificates used to connect to the grid such as for LifeCycle Manager, ION Desk, and users.
8. Reissue all host SSL keystores where there are only grid-signed SSL certificates.
9. For any SSL keystores that contain certificates signed by external Certificate authorities, import the grid root certificate into that keystore with the alias of *gridName*_cert.
10. Restart all services.
11. Import any application settings that were exported in the first step.

To reinitialize grid integrity (alternative method for LifeCycle Manager controlled grids)

1. In the Applications view, right-click on the grid and select Grid Maintenance > Manage Security.
2. Select Reinitialize Grid Integrity and click Next.
3. Select the check box to confirm the action and click Next.
4. On the Summary window, click Finish.
5. When the process is finished, restart the LifeCycle Manager client to refresh the connection between the LifeCycle Manager client and the grid.
6. This process will take care of all steps in the other procedure for re-initializing the integrity of the grid except for the step where you reissue certificates for all clients that connect to the grid. This process only reissues the certificate for the LifeCycle Manager client, so you must manually reissue the certificates for other clients such ION Desk or users.

Authentication

3

Authentication Overview

Some applications in the grid require users to be authenticated and have an established grid session to allow access to their published services. There are three ways to establish a grid session: log on using a user name and password, log on using a Lawson SSO token or using the public key infrastructure of a grid, and log on using the SSL handshake.

The SSL handshake method is built in to the framework of the grid and is used internally to ensure grid integrity and security. It is possible to provide external grid clients with the necessary key material so that a connection can be made as an authenticated user. Since this is an intrinsic mechanism, no session provider needs to be configured for clients to log on to the grid. This method of connection is used in a bootstrap situation, for example, when the grid is initially installed.

The other two methods are delegated to a pluggable architecture called the session provider. A session provider is a grid application that has been given the special right to define who is allowed to log on and to establish some sort of security context for a user. A session provider offers its services through a grid-provided interface (the SessionProvider interface hierarchy) so that other applications can use the services without having to know the implementation details.

A configured session provider is typically needed when you have applications in the grid that require users to be authenticated.

All session providers give a grid administrator a last say in the assignment process through role mapping. Through role mapping, the session provider roles are mapped to grid roles. For more information on role mapping, see "Defining Role Mappings" on page 64.

The reason for having pluggable authentication architecture is flexibility. The grid framework can never foresee all possible flavors of user storage and credential mechanisms, so it makes sense to decouple that functionality from the grid release cycle. This introduces a dilemma: the application programming interfaces used internally to create a session in the grid must be reachable for every application since a session provider is an ordinary grid application, and as an administrator you would like to have some control over how users are allowed to log in to the grid (or at least you would like to know who is allowed to perform this gatekeeper task). The grid framework solves this by requiring an application that should be allowed to create grid sessions to be registered as a session provider in that particular grid. Only an administrator can grant session provider privileges to an application, thus making this procedure secure. An application that tries to create sessions without being registered as a session provider will fail with an error report.

Grid Principals and Sessions

When a user has been authenticated, a grid principal and a session are created for that user. The grid principal is the grid's logical representation of a user and the accompanying information, such as roles. The grid and its applications can query the grid principal for role membership to make authorization decisions.

A grid session or more correctly, a grid session identifier, is what a client uses to refer to a grid principal. The session identifier lives on the client, and the grid session, with its associated grid principal, is handled by the grid. When a client needs to access a proxy interface, it sends the session ID along with the proxy call. The grid can then validate the call using the associated grid principal.

A grid session times out after a pre-configured amount of inactive time. The default value is 60 minutes, but this can be changed by editing the grid security property "Default Session Timeout". When a session either times out or is explicitly deleted, any client trying to refer to that session receives an exception back. It is up to the client to deal with that exception in a suitable manner.

Certificate-based Authentication

When using a trusted client (any client connecting with SSL--see illustration in "Certificates in the Grid" on page 7), there is no need for session identifiers to be propagated. In that case, the grid principal is intrinsic to the connection between the client and the grid and can be directly queried on the server side. Note that in this case no session is created and, therefore, there is no session timeout. The authentication is valid as long as the SSL connection stays open.

Active Sessions

All current sessions can be viewed from the Grid Management Pages, by selecting Advanced > Sessions. For each session, the principal, its roles, the origin of the call that established the session, and its remaining life are listed. It is also possible to delete sessions prematurely. Only users with the grid-admin role are allowed to view the contents.

Grid Principal Name for various Session Providers

After a successful logon, a grid principal is created to store information about the user session. The session ID is a reference to the session and this session contains the grid principal. One of the most commonly used grid principal attributes is the name of the user.

The name is created differently depending on which session provider is in use. This is important because the name is also used in grid role mappings; see "Defining Role Mappings" on page 64. Some session providers truncate the username used for logon, for example removing domain information.

This table shows how the name of the grid principal is created for the different session providers:

Session provider	Description
Windows Session Provider	The name of the grid principal is the truncated username. If a user logs on with *mydomain\user* the name of the grid principal will be "user". If a user logs on with *user.name@mydomain.com* the name of the grid principal will be "user.name".
SAML Session Provider	The name of the grid principal is retrieved from the Identity Claim included in the returned SAML token from the IdP. To configure which claim should be the Identity Claim, you can set the "sp.identity.claim.name" property for the SAML Session Provider. This property is normally set during installation. When IFS is used, the default claim value is **`http://schemas.infor.com/claims/Identity`**.
LDAP Session Provider	The name of the grid principal is set to the username used for logon. The name is set to the correct character case according to the LDAP server.

Session provider	Description
DSSO Session Provider	The DSSO Session Provider has a property called "use.identity.as.principal.name". If this property is set to false, the RMID/Actor of the user will be used as name of the grid principal. • If the RMID/Actor is *user@mycorp.com* then the name will be *user@mycorp.com*. If this property is set to true, the Identity used for logon will be used for the name, but it is truncated. • If a user logs on with *mydomain\user* the name of the grid principal will be "user". • If a user logs on with *user.name@mydomain.com* the name of the grid principal will be "user.name".

Session Provider Requirements and Selection

There are four different session providers available as grid extensions for production scenarios. The purpose of each of these session providers is to provide an authentication or validation mechanism for the grid.

Of the four session providers, only the Windows Session Provider is supported for the ION Grid installed through a Java program. For the ION Grid for LifeCycle Manager, all four session providers are supported.

In addition, to determine the most appropriate one to use, consider the following information:

Session Provider Types

Windows Session Provider

This session provider uses the same authentication mechanisms as Windows itself and provides support for NTLM and Kerberos authentication. It must be installed on a Windows 2008 R2 or Windows 2012 server belonging to the Windows domain against which it will authenticate. The Windows Session Provider supports the following authentication methods: basic authentication, NTLM, and Negotiate.

LDAP Session Provider

This session provider supports complex authentication options, including multiple domains, server fail-over options, and authentication against standalone LDAP servers. The LDAP Session Provider can be used for authenticating users to any LDAP server, including Active Directory. The LDAP Session Provider supports basic authentication using the LDAP authentication method Simple

Authentication only. This session provider requires configuration for basic setup and to take advantage of its more powerful features.

The LDAP Session Provider should not be configured to directly connect to a Lawson Security LDAP server (such as the Infor Lawson System Foundation LDAP). When authenticating against a Lawson Security system, use the DSSO Session Provider.

DSSO Session Provider

The DSSO Session Provider can authenticate against Lawson Security (in Infor Lawson System Foundation or Infor Java Framework runtime environments) using a DSSO base component. It is also used within Infor Java Framework itself, where the DSSO Session Provider communicates directly with the main Lawson Security installation. This session provider is needed if you are running Infor Smart Office in a grid, and you want to authenticate to your Infor Lawson System Foundation or Infor Java Framework runtime environment. In this scenario, the DSSO Session Provider requires the DSSO base component. For more information, see the *Distributed Single Sign-on for Lawson Smart Office Installation Guide*. The DSSO Session Provider supports basic authentication.

SAML Session Provider

The SAML Session Provider authenticates users using SAML to communicate with AD FS 2.0. User credentials are stored in AD but also synchronized to Infor Federation Services (IFS) for extended attributes (Claims) and also security role assignment (which happens in IFS). The session provider supports the following authentication methods: basic authentication and SAML 2. The SAML Session Provider implements basic authentication using WS-Trust to authenticate users to AD FS 2.0 (for active, non-browser based clients). The SAML 2 authentication method uses WS-Federation (for browser clients that can be automatically redirected).

Requirements and Selection

There are four different deployment scenarios for the session providers:

- **AD FS 2.0 and Infor Federation Services**

  In this scenario, the users are authenticated to AD FS 2.0 using the SAML protocol. Infor Federation Services is installed on the AD FS server for the extra attributes it provides and also for automating configurations. This scenario applies when Infor Ming.le™ is used with AD FS 2.0.

- **Active Directory**

  In this scenario, Active Directory is used as the user information storage, but AD FS is not used. Users are authenticated directly to the AD.

- **Lawson Security**

  In this scenario, Lawson Security is used for user authentication. The session provider used in this scenario will relay the authentication request to the configured Lawson Security System (Infor Lawson System Foundation or Infor Java Framework runtime environment). Lawson Security may store the user credentials in any LDAP or even Active Directory, but this is irrelevant from the session provider's point of view.

- **Other LDAP**

This scenario is for all other scenarios where users are stored in LDAP. The session provider authenticates the users directly to the LDAP server.

Choosing a Session Provider Based on the Scenario

An **X** in the matrix below means that the session provider supports the given scenario.

Session Provider	AD FS 2.0/IFS	Active Directory	Lawson Security	Other LDAP
Windows		-- X --		
LDAP		-- X --		-- X --
DSSO			-- X --	
SAML	-- X --			

Choosing between the LDAP Session Provider and the Windows Session Provider

If your choice is between the LDAP Session Provider and the Windows Session Provider, consider the following information:

	LDAP Session Provider 1.9	Windows Session Provider 1.9
Platform Requirements	All platforms supported by Lawson Grid 10.1.9.0 and higher, and by ION Grid 11.1.11.0 and higher.	Windows Server 2008 R2 or Windows 2012 only, where that server is part of the domain that the session provider should authenticate against. Lawson Grid 10.1.9.0 and higher, and ION Grid 11.1.10.0 and higher.
Fail-over Support	Explicit fail-over support to selected secondary servers.	Implicit fail-over support from the built-in fail-over support in Windows.
Multiple Domain Support	Explicit support for multiple domains or for just standalone LDAP servers.	
Configuration Differences	Requires configuration for basic setup and to take advantage of its more powerful features	No configuration required.

If you meet the requirements for the Windows Session Provider and do not need the explicit fail-over support or multiple domain support, use the Windows Session Provider.

System Requirements for Session Providers

The following software requirements must be met before you install session providers.

Component	Supported Version(s)	Notes
Operating System	Windows 2008 R2 Windows 2012 AIX 6.1 AIX 7.1 IBM i 6.1 IBM i 7.1 Suse Linux Enterprise Server 10SP2+ or 11SP1+	The Windows Session Provider can only be installed on Windows 2008 R2 or Windows 2012.
LifeCycle Manager	10.1.x	This is a requirement for the ION Grid for LifeCycle Manager, not the ION Grid installed via a Java program.
JVM	Oracle Java 6, update 45+ (32-bit or 64-bit)) Oracle Java 7, update 21+ (32-bit or 64-bit) IBM for i Java 6, PDF group 23+ (32-bit or 64-bit) IBM for i Java 7, PTF group 12+ (32-bit or 64-bit)	Always use the latest release at the time of installation. Maintain the Java version at the latest version with regular upgrades. Make sure that the LCM service is installed with the J9 JVM and not the classic JVM on IBM i.
ION Grid	11.1.11.0 for all 11.1.11.x grid extensions	
Infor Federation Services	10.3.2+	Required for SAML Session Provider
AD FS	2.0	Required for SAML Session Provider

Downloading the Session Providers

Download the session providers from the Infor Xtreme download page. They are available in the Infor ION Grid Extensions package.

Product name	Contains
Infor ION Grid Extensions	**Infor_ION_Grid_Extension_11.1.11.**<*minorVersionNbrs*>**.lcm** This package includes four session providers as well as the GDBC and Event Hub grid extensions.

Uploading the Session Providers to the LifeCycle Manager

Use this procedure to upload the ION Grid Extensions to the LifeCycle Manager. The ION Grid Extensions package includes the four session providers. After you have uploaded the grid extensions, you can install the session providers in an ION Grid for LifeCycle Manager. If you have an ION Grid installed through the Java installation program, the session providers available for that grid are bundled with the applications for that grid.

❑ Upload ION Grid Extensions to LifeCycle Manager

___**1** Log on to LifeCycle Manager as administrator.

___**2** Select Admin > Admin View. The Manage Products tab is displayed by default.

___**3** Click Upload and select the file **Infor_ION_Grid_Extensions**_ *version***.lcm** from the place on your client where the downloaded packages are stored.

___**4** On the Verifying package window, click Yes to accept to register the packages on the LifeCycle Manager Server.

___**5** When the task is finished, a dialog appears. Click OK.

___**6** When the dialog appears asking you if you want to update your client, click Yes.

___**7** When the update is done, a dialog appears informing that the client needs to be restarted. Click OK to restart the client.

___**8** Log in again.

Installing and Configuring the LDAP Session Provider Grid Extension

Use these procedures in install the LDAP session provider for a grid. To determine if this is the appropriate session provider to install, see "Session Provider Requirements and Selection" on page 27.

❑ The user login name

In some LDAP configurations, the user name used for login is not the same as the one that is assigned to the Grid Principal after a successful login. This depends on the overall configuration of the LDAP connections such as the selected user attribute, if any domain information is removed from the user name before login or if any advanced user search filter is used.
Some grid applications depend on being able to retrieve the original user login name from the Grid Principal. Therefore, there is a setting to provide the login name to these applications even if the Grid Principal gets another name. The ability to add the extra data to the Grid Principal was introduced in the LDAP Session Provider 1.9.20 and 1.10.9. You can set the property using the Grid Management UI and during installation. This feature might have a negative effect on performance in systems with many concurrent users. Therefore, do not enable this feature if not necessary.

❑ Install the LDAP Session Provider Grid Extension

___**1** In LifeCycle Manager, select Actions > Install Product.

___**2** From the list, select the product **Infor LDAP Session Provider** <*version*>.

Click Next.

___**3** On the Install window, select the location for the session provider. This is the grid on which the extension will be installed.

Click Next.

___**4** On the LDAP Server Configuration window, enter the host where the grid extension will be installed.

Click Next.

___**5** If you have a configuration where the login name is not the name of the Grid Principal after a successful login, on the Keep additional session data window, select Create Property Principal and add login name. See "The user login name" on page 32.

___**6** On the Summary window, verify the properties provided.

Click Finish.

___**7** Configure the LDAP Session Provider. See "Configure the LDAP Session Provider" on page 32.

❑ Configure the LDAP Session Provider

___**1** In the left pane in LifeCycle Manager, locate the LDAPSessionProvider application within the grid where you installed it.

___**2** Right-click the LDAPSessionProvider application and select Configure Session Provider.

If this is the first time that you are attempting to configure the session provider, the Server Connection window appears. If you have entered configuration information previously, the

LDAP Session Provider Editor page appears in the right pane of LifeCycle Manager. This page has several tabs where you configure different aspects of the session provider.

___**3** On the Server Connection window or Connection tab , enter the following:

Primary Server	The host name of the primary LDAP server.
Port	The port the LDAP service is listening on. Unless you have a very unique environment, leave it undefined and the correct defaults will be used (389 or 636).
Encryption method	Select "Use StartTLS extension," "Use SSL Encryption (ldaps://)," or "No encryption." The default is "Use StartTLS extension." The "Use StartTLS extension" and "Use SSL Encryption (ldaps://) methods allow password to be sent securely. Both of these use SSL/TLS protocol to secure the transmission. The main difference is that ldaps:// encrypts the entire conversation while StartTLS only encrypts the transmission of sensitive data (such as the password). This means that StartTLS is much faster and less demanding of resources. For those reasons, it is the default setting for a new connection. The certificates needed for successful communication are saved automatically by the configuration editor. Click Validate to check if the configuration editor can connect to the LDAP server. Depending on the encryption method you selected, you may need to respond to a Certificate Trust dialog box. This dialog box is displayed in different places depending on if you are configuring for StartTLS or LDAPS. When a StartTLS connection is used, the Certificate Trust dialog box is displayed when you click on Validate in the "Connection" window or tab (step 3). When LDAPS is used, the Certificate Trust dialog box is displayed when the Validate button is clicked in the "Authentication & Search Base" window or tab (see step 4). You have to select "Always trust this certificate" in order for the LDAP Session Provider to be able to connect to the LDAP server.
Secondary servers	Optionally, you can add secondary servers for fail-over purposes. For more information, see "Add a secondary server" on page 36.

Click Next on the Server Connection window or click Save on the LDAP Session Provider Editor page.

___**4** On the Authentication & Search Base window or tab, enter the following:

Username	The user name or DN to bind with. This is the user to connect to the LDAP server with and to search for users being validated. It must be either a fully qualified name in the form **`"cn=User, ou=Users,dc=corp,dc=example,dc=com"`**, or in the case of an Active Directory environment, **`"User@corp.example.com"`** will also work.

Password	The password to bind with. Click Validate to confirm that the user name and password are correct.
Search base	LDAP location to be added to the connection URL in searches, for example, **dc=corp,dc=example,dc=com**. You can click Lookup to list all possible bases.

Click Next on the Authentication & Search Base window or click Save on the LDAP Session Provider Editor page.

___**5** On the User Element Mapping window or tab, configure the user element mapping. There are two different configuration modes available. The Simple Search offers basic configuration that is enough in most cases. With the Advanced Search mode, a complete user search filter can be configured. Most of the configuration elements are common for both configuration modes.

Enter the following:

Base Locations	Base locations in the LDAP where the users are found, relative to the search base (for example, "ou=Users"). Multiple base locations can be added, if you have users located in more than one part of the LDAP tree. Click on Add on the right and browse to the preferred part of the LDAP tree. Select a Base DN location from the list and click on Remove to delete the Base DN location from the list.
User Scope	Select Sub-tree or leave the check box clear. If Sub-tree is selected, the search is from the Base DN and down, rather than just in the base locations. In most cases, Sub-tree should be selected. The User Scope setting is identical for all configured base locations.
User ID Attribute	LDAP attribute containing user id. Default value: **cn**. The default is used if no value is entered. The User ID Attribute setting is identical for all configured base locations.
Simple Search	For the simple search, only the object class the users belong to needs to be configured.
Object Class	LDAP class for user objects. Default value: **user**. The default is used if no value is entered. The Object Class setting is identical for all configured base locations.
Advanced Search	With an advanced search, the complete user search filter can be supplied. When switching from simple to advanced, a proposed example filter is provided based on the values in Object Class and the User ID Attribute.

Filter	A user search filter can be entered in the enabled text field when Advanced Search is selected. If a filter is entered, the Object Class property is not used any longer and that field is disabled. Enter %USER% in the filter where substitution should take place for the name of the user who is logging in or for the user name being searched for. A typical filter can look like this: **`(&(objectClass=user)(sAMAccountName=%USER%))`** The Filter setting is identical for all configured base locations.
Strip domain	If this option is selected, the provided domain information in the user names will be removed before the login is made. Do not select this option in configurations where the domain information must be kept for login.

Click Validate to confirm that a search can return a list of users. The validation will test each of the provided base locations provided above. If no base location is provided, the relative search base is used for validation. The validation will show an example result for each base location

Click Next on the User Element Mapping window or click Save on the LDAP Session Provider Editor page.

___**6** On the Group Element Mapping window or tab, enter the following:

Base Locations	Base location in the LDAP where the groups are found, relative to the search base ("ou=Groups"). For performance reasons, it is best to specify the most specific Base DN possible. This is because the search must search and map all groups under the Base DN in order to find the groups a user is a member of. Note that the LDAP session provider can only find groups that users are direct members of. You therefore cannot use groups within groups. You can add multiple different Base DNs if you have groups located in more than one part of the LDAP tree. Click on Add on the right and browse to the preferred part of the LDAP tree. Select a Base DN from the list and click on Remove to delete the Base DN from the list.
Object Class	LDAP class for group objects. Default value: **`group`**. The default is used if no value is entered. The Object Class setting is identical for all configured base locations.
Group Member Attribute	LDAP attribute containing group id. Default value: **`member`**. The default is used if no value is entered. The Group Member Attribute setting is identical for all configured base locations.
Group Scope	Select **`Subtree`** or leave the check box clear. The default is to leave the check box clear. If **`Subtree`** is selected, the search is from the Base DN and down, rather than just in the Base DN. In most cases, **`Subtree`** should be selected. The Group Scope setting is identical for all configured base locations.

Click Validate to confirm that a search can return groups. The validation will test each of the provided group mapping base locations provided above. If no base location is provided, the relative search base is used for validation. The validation will show the result for each base location.

Click Finish on the Group Element Mapping window and then click Save on the LDAP Session Provider Editor page.

___**7** After you have configured the LDAP session provider, you can set up role mapping for securing users.

❑ Add a secondary server

You can add secondary LDAP servers to your configuration for fail-over purposes. The implementation checks each call to the LDAP server (that fails) and looks for some specific exceptions/errors. When one of the known errors is seen, it is interpreted as a failed server and the session provider switches to the next server in the list. The switch is done in a round-robin fashion, and the state is not saved between restarts of the session provider. Therefore, you must make sure to keep the primary server first in the list.
Note that if a fail-over occurs during an attempted logon, that logon will fail. The new server will be used by the next logon attempt. When a switch happens, an INFO message similar to the following is logged in the SessionProvider log:

```
2013-03-04 08:12:38,525 INFO SessionProvider SessionProvider: Switching server from
sestw426.corpnet.infor.com to ldapemea.corpnet.infor.com
```

The configured servers, as well as the currently active server, can be seen by selecting the session provider in the LifeCycle Manager, and selecting Manage Application. If fail-over is configured, a list of the servers is shown, with an asterisk ("*") next to the currently active server

___**1** Locate the LDAPSessionProvider application for the grid in the left pane in LifeCycle Manager.

___**2** Right-click the LDAPSessionProvider application and select Configure Session Provider.

___**3** On the Connection tab, click the Add... button by the Secondary server list field.

___**4** Enter the address to the secondary server you want to add and click OK. After you click OK, a check is made to see if it is possible to connect to the server.

___**5** Click Add... again if you want to add more secondary servers.

___**6** When you are finished adding secondary servers, click Save.

___**7** Switch to the Authentication & Search Base tab and click Validate. This will validate that the username and password are valid on all servers. You might also get additional certificate dialogs if you use any of the SSL-based encryption methods.

❑ Add additional domains

If you have users in multiple domains, you can add those domains to the configuration. The session provider will look for users in all domains simultaneously. Should a username exist in more than one domain, the logon will fail. The reason for this is that the session provider cannot know if the two users are identical, or if they should be treated differently.

___**1** Locate the LDAPSessionProvider application for the grid in the left pane in LifeCycle Manager.

___2 Right-click the LDAPSessionProvider application and select Configure Session Provider.

___3 Click the Add Domain button in the upper right corner.

___4 When you are presented with the Server connection window, enter values just as if you were configuring the session provider, except with values appropriate for the new domain. For information on the session provider configuration fields, see "Configure the LDAP Session Provider" on page 32.

Installing the DSSO Session Provider

Use this procedure to install the DSSO session provider. To determine if this is the appropriate session provider to install, see "Session Provider Requirements and Selection" on page 27.

Important: For Lawson Security servers using Kerberos authentication, you must use the DSSO Session Provider 1.3. For installation instructions, see "Install the DSSO Session Provider 1.3 in a grid" on page 39.

For all other scenarios, use the DSSO Session Provider 2.0.

The installation for the DSSO Session Provider 2.0 is divided into three phases or tasks. The first phase and the last phase are performed using the DSSO Session Provider LifeCycle Manager plugin. The second phase is performed using one of the LifeCycle Manager plugins that can install a DSSO instance (minimum versions: DSSO 9.0.2.3.14 and DSP 10.0.1).

1 "Install a new grid router for the DSSO Session Provider 2.0" on page 37

2 "Install DSSO using either the DSSO or the DSP LifeCycle Manager package" on page 38

3 "Install the DSSO Session Provider 2.0 in a grid" on page 39

Before you start Before you can install the DSSO Session Provider 2.0, you must have an installation package for DSSO uploaded on the LifeCycle Manager server.

The DSSO Session Provider 2.x is not backwards compatible and an upgrade is not possible from previous versions. If DSSO Session Provider 1.x is already installed, this installation must be removed prior to the new installation as well as the DSSO instance for that session provider.

Note: When you install the DSSO Session Provider 2.0, you must install the grid router, the DSSO instance, and the DSSO Session Provider on the same grid host.

❑ Install a new grid router for the DSSO Session Provider 2.0

___1 In LifeCycle Manager, select Actions > Install Product.

___2 From the list, select the **DSSO Session Provider** *version* product with the description "Create a new Grid Router adapted for DSSO use."

___3 Select the host to install the router on and click Next.

___**4** Provide the external FQDN and the ports for the router.

Both the Lawson Security Server and the clients must be able to resolve and reach the provided FQDN.

Write down the provided FQDN and ports since they must be provided when you install DSSO. For more information, see "Install DSSO using either the DSSO or the DSP LifeCycle Manager package" on page 38.

Click Next.

___**5** On the Summary window, click Finish.

❑ Install DSSO using either the DSSO or the DSP LifeCycle Manager package

___**1** If you are installing DSSO using the DSP LifeCycle Manager package, follow the instructions in the *Distributable Security Package Installation Guide*. Review the section for Infor Smart Office or ION Enterprise Search.

a In the installation step "DSSO Instance," the provided instance name will be the name of the DSSO Service you will provide when completing the third task for installing the DSSO Session Provider. See "Install the DSSO Session Provider 2.0 in a grid" on page 39.

b In the installation step "Web Application Server," select the application server type of "Manual deployment but create service." For the FQDN, enter the external FQDN of the DSSO router you provided in the first task of installing the DSSO Session Provider. For the HTTP/HTTPS ports, use the ports defined in the first task of installing the DSSO Session Provider. See "Install a new grid router for the DSSO Session Provider 2.0" on page 37.

c When the DSSO installation is finished, create new identities for the users that should have access to the newly created service. These identities must be on the newly created service. In addition, these users must have identities on the primary domain service (usually SSOP) as those identities are used for authentication for the DSSO Session Provider.

___**2** If you are installing DSSO using the DSSO LifeCycle Manager package, follow the instructions in the *Distributed Single Sign-on for Lawson Smart Office Installation Guide* (available on the download page at Technology > Enterprise Search > Shared Security Platform All supported platforms) or the *Lawson Enterprise Search for LifeCycle Manager Installation Guide* (available in the Lawson Enterprise Search infocenter or the download page at Technology > Enterprise Search > Lawson Enterprise Search VMware ESX). Review the section for Lawson Smart Office or Lawson Enterprise Search.

a In the installation step "Create DSSO Service," the provided Unique Service name will be the name of the DSSO Service you will provide when completing the third task for installing the DSSO Session Provider. See "Install the DSSO Session Provider 2.0 in a grid" on page 39. Select the check box for "Create DSSO Service instance" and select "Create new service" in the "Installation type" drop-down box.

b In the installation step "Web Application Server," select the application server type of "No App server or manual configuration." For the "Web frontend server FQDN," enter the

external FQDN of the DSSO Router provided in the first task of installing the DSSO Session Provider. For the HTTP/HTTPS ports, use the ports defined in the first task of installing the DSSO Session Provider. See "Install a new grid router for the DSSO Session Provider 2.0" on page 37.

c When the DSSO installation is finished, create new identities for the users that should have access to the newly created service. These identities must be on the newly created service. In addition, these users must have identities on the primary domain service (usually SSOP) as those identities are used for authentication for the DSSO Session Provider.

❑ Install the DSSO Session Provider 2.0 in a grid

___1 In LifeCycle Manager, select Actions > Install Product.

___2 From the list, select the **Infor DSSO Session Provider** <*version*> product with the description "Deploy the DSSO Session Provider."

___3 On the Install window, select the location for the DSSO Session Provider. This is the grid on which the DSSO Session Provider will be installed.

Click Next.

___4 On the DSSO Base installation window, select the DSSO base installation created in the "Install DSSO" task and click Next.

___5 On the Lawson Environment window, consider the following fields and click Next when you are done:

Connected to Environment	This read-only field displays the name of the Lawson Environment based on the DSSO base installation value you entered earlier.
Authentication service name	Enter the name of the authentication service as defined in the "Install DSSO" task. That is, if you are installing DSSO using the DSP LifeCycle Manager package, this is the name of the DSSO Service you provided in the "DSSO Instance" step. If you are installing DSSO using the DSSO LifeCycle Manager package, this is the name of the DSSO Service you provided in the "Create DSSO Service" step.

___6 On the Grid properties window, select which router to use. This should be the DSSO router created in the "Install a new Grid router for the DSSO Session Provider."

___7 On the Summary window, click Finish.

___8 After you have installed the DSSO session provider, you can set up role mapping for securing users.

❑ Install the DSSO Session Provider 1.3 in a grid

Before you start Before you can install the DSSO Session Provider 1.3, you must have already installed the DSSO base components. For more information, see the *Distributed Single Sign-on for Lawson Smart Office Installation Guide*.

The minimum grid version for using the DSSO Session Provider 1.3 is 10.1.9.0.

___**1** In LifeCycle Manager, select Actions > Install Product.

___**2** From the list, select the product **DSSOSessionProvider**_*version* product.

___**3** On the Install window, select the location for the DSSO Session Provider. This is the grid on which the DSSO Session Provider will be installed.

Click Next.

___**4** On the DSSO Base installation window, select the DSSO base installation and click Next.

___**5** On the Lawson Environment window, consider the following fields and click Next when you are done:

Connected to Environment	This read-only field displays the name of the Lawson Environment based on the DSSO base installation value you entered earlier.
Authentication service name	Enter the name of the authentication service. (The default value is the primary service for the Lawson Environment). You can check the name of the primary service by accessing the following URL: **`http://`***`servername`***`:`***`port`***`/ssoconfig/SSOCfgInfoServlet`**.

___**6** On the Grid properties window, enter a name for the DSSO Session Provider installation and select the JDK version (either 32-bit or 64-bit).

___**7** On the Summary window, click Finish.

___**8** After you have installed the DSSO Session Provider, you can set up role mapping for securing users.

Troubleshooting the DSSO Session provider

Troubleshooting the DSSO Session Provider 1.3.x

Things to consider

1. The DSSO Session Provider does support reusing the primary authenticating service: SSOP, SSOPV2, or LSS. You can create a new form-based DSSO Service and use that. If you do this, then also create Identities for all users that must be able to authenticate.
2. When installing DSSO, the Remote OS Service must be created, and the Actor/RMID and Identity for the service account for the Grid Agent service. This is normally the "NT AUTHORITY\Local Service".
3. Users authenticating must have an Actor/RMID and an Identity for the service configured in DSSO Session Provider.
4. Always turn on Debug and Trace logging for the DSSO Session Provider to get detailed messages about the possible problem. The logging generates extensive debug output; therefore turn off debugging and tracing after finishing when the issue is solved.

5 If The DSSO Session Provider must authenticate to a Lawson Security environment where a LSF is coupled with Landmark, the LMRK system is the primary authenticator. The DSSO installation that the DSSO Session Provider is connected to must be for the LMRK system. The DSSO Session Provider must be configured to use a LMRK service, normally SSOPV2, for authentication.

Possible problems

Issue: OS account not configured for authentication

This message is found in the DSSO SessionProvider log:

```
WARN DSSOSessionProvider DSSOSessionProvider: Authentication exception
com.lawson.security.authen.SecurityAuthenException: Your OS account has not been configured
 for authentication. at
com.lawson.security.authen.RemoteOSAuthenticatorImpl.getPrimordialContext(RemoteOSAuthenticatorImpl.java:227)
at provider.DSSOSessionProvider.validatePassword(DSSOSessionProvider.java:193) at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.logon(AbstractSessionProviderBase.java:175)
 at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.logon(AbstractSessionProviderBase.java:209)
 at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.authenticate(AbstractSessionProviderBase.java:287)
```

Remedy:

This issue means that the Account that the Grid Agent service is running as does not have an identity or RMID/Actor in the Lawson Security server. The DSSO installer should create rmids/actors and identities for Local Service and Local System during installation.

If the Windows Session Provider has ever been installed on the same grid host as the DSSO Session Provider is being installed, the Grid Agent service account is changed from Local Service to Network Service.

To resolve this issue, complete one of these steps:

- Change the service account of the Grid Agent to Local Service.
- Create an rmid/actor and an identity for the used service, probably SSOP or SSOPV2, for Network Service.

Issue: OS Service has an identity but no Actor/RMID

This message can be found in the DSSO Session Provider log when Debug is enabled:

```
com.lawson.security.authen.SecurityAuthenException: Bad id. Is null or empty.
 at
com.lawson.security.authen.LawsonUserContextImpl._setAuthenticatedActorObject(LawsonUserContextImpl.java:338)

 at com.lawson.security.authen.LawsonUserContextImpl.<init>(LawsonUserContextImpl.java:84)
 at com.lawson.security.authen.LawsonOSUserContextImpl.<init>(LawsonOSUserContextImpl.java:21)
 at
com.lawson.security.authen.RemoteOSAuthenticatorImpl.getPrimordialContext(RemoteOSAuthenticatorImpl.java:222)
```

The message can also be "**`Your OS account has not been configured for authentication`**."

To solve this issue, ensure that the OS Service is correctly configured. Also ensure that the service account for the Grid Agent on the DSSO Session Provider host has an Identity and an Actor/RMID.

Remedy:

Create the OS Service in the Lawson Security environment. You can manually create the OS Service on an LSF server or a Landmark server. See the appropriate documentation for the used system.

Issue: Jar file version mismatch

The jar files used in the DSSO instance must match the version in the Lawson Security environment to which it is connected. This error message is an indication that the jar files do not match up.

```
WARN DSSOSessionProvider DSSOSessionProvider: Authentication exception
com.lawson.security.server.LawsonNetException:
com.lawson.security.server.NetMessages.readSerializedObject(): Could not deserialize object
 data: [IOException] : com.lawson.security.authen.AbstractLawsonService; local class
incompatible: stream classdesc serialVersionUID = 1, local class serialVersionUID =
-5215695224618207798
at com.lawson.security.server.events.BaseEvent.readSerializedObject(BaseEvent.java:1219)
```

Remedy: Update DSSO instance

Update the DSSO instance to have the same jar file versions as the Lawson Security server.

Troubleshooting the DSSO Session Provider 2.x

The DSSO Session Provider 2.x differs drastically from the 1.x versions. By using the SSO servlets, like any other DSSO application does, the 2.x version takes a more regular DSSO approach than the earlier version.

Things to consider

1. The DSSO Session Provider does not support reusing the primary authenticating service (SSOP, SSOPV2 or LSS) since it now is using the session migration techniques used by DSSO. A DSSO Service configured for the DSSO Router must be used.
2. The used DSSO Service must have the HTTPURL, HTTPSURL, and SERVICEURL pointing to the DSSO Router like this:
   a. "**<HTTPURL>http://[DSSO Router external FQDN]:[DSSO Router http port]/sso/SSOServlet</HTTPURL>**"
   b. "**<HTTPSURL>https://[DSSO Router external FQDN]:[DSSO Router https port]/sso/SSOServlet</HTTPSURL>**"
   c. "**<SERVICEURL>http://[DSSO Router external FQDN]:[DSSO Router http port]/sso/SSOServlet</ SERVICEURL >**" (if HTTP Only is selected as Assertion Protocol in the DSSO installation)

The easiest way to realize this, is to specify the correct values when performing the DSSO installation for the DSSO Session Provider.

3 When installing DSSO, the Remote OS Service must be created, and the Actor/RMID and Identity for the service account for the Grid Agent service. This is normally the "NT AUTHORITY\Local Service".

4 Always turn on Debug and Trace logging for the DSSO Session Provider to get detailed messages about the possible problem. The logging generates extensive debug output; therefore turn off debugging and tracing after finishing when the issue is solved.

Possible problems

Issue: OS account not configured for authentication

This message is found in the DSSO SessionProvider log:

```
WARN DSSOSessionProvider DSSOSessionProvider: Authentication exception
com.lawson.security.authen.SecurityAuthenException: Your OS account has not been configured
 for authentication. at
com.lawson.security.authen.RemoteOSAuthenticatorImpl.getPrimordialContext(RemoteOSAuthenticatorImpl.java:227)
at provider.DSSOSessionProvider.validatePassword(DSSOSessionProvider.java:193) at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.logon(AbstractSessionProviderBase.java:175)
 at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.logon(AbstractSessionProviderBase.java:209)
 at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.authenticate(AbstractSessionProviderBase.java:287)
```

Remedy:

This issue means that the Account that the Grid Agent service is running as does not have an identity or RMID/Actor in the Lawson Security server. The DSSO installer should create rmids/actors and identities for Local Service and Local System during installation.

If the Windows Session Provider has ever been installed on the same grid host as the DSSO Session Provider is being installed, the Grid Agent service account is changed from Local Service to Network Service.

To resolve this issue, complete one of these steps:

- Change the service account of the Grid Agent to Local Service.
- Create an rmid/actor and an identity for the used service, probably SSOP or SSOPV2, for Network Service.

OS Service has an identity but no Actor/RMID

This message can be found in the DSSO Session Provider log when Debug is enabled:

```
WARN DSSOSessionProvider    DSSOPrincipalUtils: Could not create grid principal
com.lawson.security.authen.SecurityAuthenException: Bad id. Is null or empty.
 at
com.lawson.security.authen.LawsonUserContextImpl._setAuthenticatedActorObject(LawsonUserContextImpl.java:338)

 at com.lawson.security.authen.LawsonUserContextImpl.<init>(LawsonUserContextImpl.java:84)
 at com.lawson.security.authen.LawsonOSUserContextImpl.<init>(LawsonOSUserContextImpl.java:21)
```

```
 at
com.lawson.security.authen.RemoteOSAuthenticatorImpl.getPrimordialContext(RemoteOSAuthenticatorImpl.java:222)
```

Remedy:

To solve this issue, ensure that the OS Service is correctly configured. Also ensure that the service account for the Grid Agent on the DSSO Session Provider host has an Identity and an Actor/RMID.

You can manually create the OS Service and any missing Actor/RMID and Identities on an LSF server or a Landmark server. See the appropriate documentation for the used system.

DSSO Service is configured incorrectly to use the wrong FQDN or ports

If the DSSO Service was not configured correctly during DSSO installation, the session migration from the primary servlet to the secondary (DSSO) servlet fails. If this is the case, the log file shows the following for a login attempt, if debug logging is activated for the DSSO Session Provider:

1. The DSSO Session Provider debug logs show that login to the primary servlet was successful. The log contains the "**`SSO_STATUS returned: LoginSuccessful`**" message. This does not imply that a grid session has been established.
2. The session migration failed from the primary server to the DSSO Session Provider. The log contains the "**`_action returned:MIGRATESESSION`**" message and a GeneralDSSOHttpclientException with the "**`Error logging in`**" and "**`Connection to http://<fqdn>:<port> refused`**" messages.

```
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: http://LSFserver:80/sso/SSOServlet returned:
 HTTP/1.1 302 Found
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: SSO_STATUS returned: LoginSuccessful
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: _action returned: MIGRATESESSION
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: ACTOR returned: myactor
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: LOGIN_IDENTKEY returned: User: myidentity
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: SSO_DOMAIN returned: DefaultSSODomain
DEBUG DSSOSessionProvider   DSSOSessionProvider: Error logging in
com.infor.security.dsso.httpclient.exception.GeneralDSSOHttpclientException:
org.apache.http.conn.HttpHostConnectException: Connection to http:// LSFserver:80 refused
 at
com.infor.security.dsso.httpclient.DSSOServerConnection.login(DSSOServerConnection.java:198)

 at com.infor.security.dsso.httpclient.DSSOHttpClient.login(DSSOHttpClient.java:224)
 at provider.DSSOSessionProvider.doValidatePasswordForSession(DSSOSessionProvider.java:283)

 at provider.DSSOSessionProvider.validatePasswordForSession(DSSOSessionProvider.java:227)
 at provider.DSSOSessionProvider.logon(DSSOSessionProvider.java:206)
 at
com.lawson.grid.proxy.access.AbstractSessionProviderBase.logon(AbstractSessionProviderBase.java:209)
Caused by: org.apache.http.conn.HttpHostConnectException: Connection to http:// LSFserver:80
 refuse
```

Remedy:

To fix this, ensure that the correct HTTPURL, HTTPSURL, and SERVICEURL are configured. See For more information, see "Things to consider" on page 42..

Primary service configured for the DSSO Session Provider

The primary authenticating service (SSOP, SSOPV2, or LSS) cannot be referenced by the DSSO Session Provider 2.x grid property server.servicename. The issue can be detected in a debug-enabled log file for a login attempt:

1. The DSSO Session Provider debug log shows that login was successful. The log contains the "**`SSO_STATUS returned: LoginSuccessful`**" message. This does not imply that a grid session has been established.
2. After the first message, there is no indication that the session was migrated to the DSSO Session Provider. In other words, the "**`LoginSuccessful`**" message is NOT followed by the "**`_action returned:MIGRATESESSION`**" message.

```
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter:
http://LSFServer:80/sso/SSOServlet?action=LOGINUSER&ssoOrigUrl=https%3A%2F%2Fgridhost%3A19008%2Fauth%2FAuthenticationServlet&_TK=TOD-UI&serviceName=SSOP
 returned: HTTP/1.1 200 OK
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: SSO_STATUS returned: Login
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: http://LSFServer:80/sso/SSOServlet returned:
 HTTP/1.1 302 Found
DEBUG DSSOSessionProvider DSSOHttpLoggerAdapter: SSO_STATUS returned: LoginSuccessful
DEBUG DSSOSessionProvider   DSSOSessionProvider: Error logging in
com.infor.security.dsso.httpclient.exception.GeneralDSSOHttpclientException:
org.apache.http.conn.HttpHostConnectException: Connection to https://Gridhost:19008 refused
```

Remedy:

Change the used service in the DSSO Session Provider to a correctly created form based service for the DSSO Session Provider Router. See For more information, see "Things to consider" on page 42..

DSSO instance version mismatch with Security Environment

The DSSO instance to which the DSSO Session Provider is connected must have binaries and settings that match the Lawson Security environment it is using. The DSSO instance must have a matching .ssotruststore file in the LASYSDIR folder. If the .ssotruststore file does not match, a "**`signature check failed`**" exception occurs. If this happens, update the DSSO instance and ensure that the **LASYSDIR/.ssotruststore** file is in sync.

```
WARN DSSOSessionProvider               JETTY: unavailable
com.lawson.security.authen.SecurityAuthenException: Failed to initialize authentication
layer. Cause Connection error (lsf.infor.com, null). Cause: {2}.
Stack Trace :
com.lawson.security.authen.SecurityAuthenException: Connection error (lsf.infor.com, null).
 Cause: {2}. at
com.lawson.security.authen.LawsonAuthentication.initClientAuthenDatThroughSSL(LawsonAuthentication.java:385)
 at
com.lawson.security.authen.LawsonAuthentication.initClientAuthenDat(LawsonAuthentication.java:247)
 at com.lawson.security.authen.LawsonAuthentication.remoteInit(LawsonAuthentication.java:2451)
 at
com.lawson.security.authen.LawsonAuthentication.initializeForTenant(LawsonAuthentication.java:184)
 at
com.lawson.security.authen.LawsonAuthentication.initializeForTenant(LawsonAuthentication.java:159)
 at com.lawson.security.authen.LawsonAuthentication.initialize(LawsonAuthentication.java:113)
 at com.lawson.security.authen.SSOCfgInfoServlet.init(SSOCfgInfoServlet.java:88) at
javax.servlet.GenericServlet.init(GenericServlet.java:241)

Caused by: com.lawson.security.authen.SecurityAuthenException:
com.lawson.security.authen.AuthenMessages.Got exception while reading from connection
5ac062e[SSL_NULL_WITH_NULL_NULL:
Socket[addr=lsf.infor.com/127.0.0.1,port=12345,localport=51957]].
Stack Trace :
```

```
com.lawson.security.server.LawsonNetException: Got exception while reading from connection
 5ac062e[SSL_NULL_WITH_NULL_NULL:
Socket[addr=lsf.infor.com/127.0.0.1,port=12345,localport=51957]].

Caused by: javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException:
 PKIX path validation failed: java.security.cert.CertPathValidatorException: signature check
 failed
```

Remedy:

Ensure that the DSSO instance used is in sync with the Lawson Security environment to which it is connected. The binaries, settings, and truststore file must be in sync.

Issue: Jar file version mismatch

The jar files used in the DSSO instance must match the version in the Lawson Security environment to which it is connected. This error message is an indication that the jar files do not match up:

```
WARN DSSOSessionProvider DSSOSessionProvider: Authentication exception
com.lawson.security.server.LawsonNetException:
com.lawson.security.server.NetMessages.readSerializedObject(): Could not deserialize object
 data: [IOException] : com.lawson.security.authen.AbstractLawsonService; local class
incompatible: stream classdesc serialVersionUID = 1, local class serialVersionUID =
-5215695224618207798
at com.lawson.security.server.events.BaseEvent.readSerializedObject(BaseEvent.java:1219)
```

Remedy: Update DSSO instance

Update the DSSO instance to have the same jar file versions as the Lawson Security server.

Installing the Windows Session Provider

Use this procedure to install the Windows Session Provider grid extension. You can install this session provider only in Lawson Grid 10.1.9.0 or higher or Infor ION Grid 11.1.10.0 or higher on a Windows server with a Windows domain.

To determine if this is the appropriate session provider to install, see "Session Provider Requirements and Selection" on page 27.

❑ Install the Windows session provider in a grid

___**1** In LifeCycle Manager, select Actions > Install Product.

___**2** From the list, select the product **Infor Windows Session Provider** *version*.

Click Next.

___**3** On the Install window, select the location for the Windows Session Provider. This is the grid on which the Windows Session Provider will be installed.

Click Next.

___**4** On the Summary window, click Finish.

___**5** After you have installed the Windows session provider, you can set up role mapping for securing users.

Installing and Configuring the SAML Session Provider

Use this procedure to install the SAML Session Provider grid extension. The SAML Session Provider should only be deployed on a single host and started in a single node.

Before you start If you want to use the SAML Session Provider, your system must meet the following requirements:

- The browser used must support Integrated Windows Authentication (IWA).
- AD FS 2.0 is used as the Identity Provider (IdP).
- Infor Federation Services (minimum version 10.3.2+) is installed on the AD FS server.
- You have a domain account that is an IFSApplicationAdmin and AttributeServiceCaller in the IFS application. Preferably this account should not expire since the SAML SP will use that account configuration for role lookup in IFS during logins.
- In AD FS the Endpoint "/adfs/services/trust/13/usernamemixed" for WS-Trust 1.3 is both Enabled and Proxy Enabled.
- You have added security roles in IFS (Manage > Master Data, double-click on Security Role) to the grid roles (grid-admin, grid-poweruser, grid-user, grid-runas) and the SAML SP test role (TestRole).

Note: Due to third-party requirements, the SAML Session Provider only supports Java 1.6.x. After installing the SAML Session Provider, ensure that the Java version for the SAML Session Provider is Java 1.6.x. Otherwise, update the Java Executable grid property for the SAML Session Provider binding. Restart the node after updating the Java version. For more information, see the *Infor ION Grid Administration Guide*.

❑ Install the SAML Session Provider in a grid using LifeCycle Manager

___**1** In LifeCycle Manager, select Actions > Install Product.

___**2** From the list, select the product **Infor SAML Session Provider** <*version*>.

Click Next.

___**3** On the Host selection window, select the grid host you want to deploy the SAML Session Provider to, and click Next.

___**4** If a SAML router already exists, you will be asked if you want to reuse that router. If no SAML router exists, on the Router properties window, define the properties for the router to be used by the session provider and click Next:

External address The external address for the router.

IP Address	The external IP address of the router.
Http port	The HTTP port for the router. The installation provides the next highest available ports as a suggestion for this field and the next field.
Https port	The HTTPS port for the router.

___**5** On the Session Provider Properties window, define the following and click Next:

Service Provider Entity ID	Provide a service provider entity ID. The recommended format is the fully qualified domain name concatenated with the HTTPS port, for example, **`acme.corp.com_61008`**. This ID will be configured in the IdP (and IFS if used).
IdP FQDN	The fully qualified domain name of the AD FS.
IdP http port	The HTTP port of the AD FS endpoint.
IdP https port	The SSL port of the AD FS endpoint.
Metadata URI	Provide the URI to the federation metadata. The default AD FS 2.0 value is "**`/FederationMetadata/2007-06/FederationMetadata.xml`**". If AD FS is used as the IdP, the URI can be found by looking in the "AD FS 2.0 Management." Open the Service folder and look in the Endpoints folder. Check the Metadata point of the screen and the "Metadata" type.

After you click Next, the installer will get the SSL certificates from the AD FS server and you will have to confirm them before continuing. The installer will retrieve the AD FS metadata and parse it for suggested values for a later installation step.

___**6** On the IFS Properties window, define the following and click Next:

Use IFS	Select this check box if it is not already selected.
IFS base URL	Provide the base URL for IFS. The suggested URL should be correct for a standard installed IFS server. If you change this value, do not include more of the URL than the virtual directory of IFS. The default value is **`http://`***`IFSServer`***`:`***`port`***`/IFSServices`**. Do not provide an HTTPS URL if that is not a defined endpoint in the IFS ConfigurationService WSDL.
Username	Provide the name for a domain user that belongs to the IFSApplicationAdmin and AttributeServiceCaller roles in IFS so that a new application can be created in IFS. The username MUST be in the domain\uid format.
Password	Provide the password for the domain user from the previous field.

After you click Next, the EntityID defined previously is validated against IFS and if the EntitiyID already exists as an application in AD FS, you will have to confirm that you want to overwrite the existing application in IFS.

___**7** Review the fields on the SAML Properties window and click Next. These properties are used by the SAML session provider when talking to the IdP and also define the endpoints the SAML Session Provider will provide for logging in and logging out. The suggested values are based on the AD FS metadata provided in previous steps. Do not change them if you are not sure of what your are doing.

___**8** Review the values on the Summary window and click Finish to start the installation.

❑ Install the SAML Session Provider in a grid manually

To install the SAML Session Provider manually, you must set up a properties file containing the deployment profile configuration data. This procedure also requires Grid version 10.1.10.0+ to enable the passing of the properties file containing the configuration data to the deployment profile.
There are 18 possible properties to configure for the manual SAML Session Provider deployment. Only eight of those are required. The other 10 will get their default value if omitted in the properties file.

Property	Required?	Description
routerFqdn	Yes	SAML-enabled grid router external FQDN.
routerIP	Yes	IP address for external access to the SAML router.
idpFqdn	No	IFS/ADFS server FQDN.
useIFS	No	Dictates if IFS setup should be performed or not. If set to **`false`**, all IFS/ADFS configuration has to be done manually. Default is **`true.`**
IFSUser	Yes, if useIFS is true or if useIFS is not defined	Username for an IFS Administrator. This user must be member of the IFSApplicationAdmin and the AttributeServiceCaller Security Roles in IFS.
IFSPass	Yes, if useIFS is true or if useIFS is not defined	The password for the IFSUser.
routerHttpPort	Yes	HTTP port to the SAML router.
routerHttpsPort	Yes	SSL port to the SAML router.
idpHttp	No	HTTP port to ADFS/IFS. Default is **80**. IFS installation guide recommends changing the port.
idpHttps	No	SSL port to ADFS/IFS. Default is **443**. IFS installation guide recommends changing the port.

Property	Required?	Description
idpUri	No	The URI to the Federation metadata xml file of ADFS. The value can be found in the AD FS 2.0 management tool under Service > Endpoints. At the bottom are the Metadata links. Default is **`/FederationMetadata/2007-06/FederationMetadata.xml`**.
idpMultiTenant	No	A boolean property to indicate if the session provider should operate in multi-tenant mode. Default is **`false.`**
ifsCfgsvcUrl	No	The virtual directory of the IFS web application on the idpFqdn. Default is **`IFSServices`**.
SignAssertions	No	Requests the IdP to sign assertions. Default is **`true`**.
identityClaimName	No	Provides which claim in an assertion to be used to decide the identity. Default is **`http://schemas.infor.com/claims/Identity`**.
assertionTimeout	No	Provides the timeout in seconds for an assertion. Default is **`300`**.
nameidFormat	No	Provides the NameIDFormat used for WS-Federation. Default is **`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`**.
IFSAppType	No	Decides which IFS application type to create. If IFS version 10.3+ is used, then use **`GRID`**. For earlier versions, use **`SAMLP`**. Default is **`GRID`**.

___1 Access the Configuration Manager for the grid and log on as a grid-admin.

___2 Click Routers.

___3 If there already is a SAML router, note the external FQDN, HTTP and HTTPS ports, and external IP address, and put these into the configuration data file. Ensure that the SAML router supports the saml2 authentication method for both HTTP and HTTPS. Also remember the host it is running on for the configuration in step 4b below.

___4 If there is no SAML router, add one:

a Click Add Router.

b Enter the following on the Router window:

Name	Enter **`SAML Router`**.
Host	Select the host where the router should run.
External Address	Provide the external FQDN for the host. This FQDN must be resolvable and reachable from both the IFS server and the connecting clients.

HTTP	On the Port tab, select an HTTP port. It is recommended to use the next port in line after the other router. On the WWW Authentications Methods tab, select **saml2**.
HTTPS	Select an HTTPS port. Recommended is to use the next port in line after the HTTP port. On the WWW Authentications Methods tab, select **saml2**.

___5 Click Add.

___6 Enter the router properties into the configuration data file.

___7 Prepare the rest of the configuration data in the file to be used during deployment. See "Deployment profile configuration data" on page 25 for information about the properties.

___8 Navigate back to the Configuration Manager home page.

___9 Click Applications.

___10 Click Install New Application.

___11 On the Select Application tab, if the SAMLSessionProvider is not available in the list, click Upload. Browse to the gar file and click Upload.

___12 When the SAMLSessionProvider is available on the Select Application tab, select it and click Next.

___13 On the Install Options tab, ensure that the name is SAMLSessionProvider and that the selected deployment profile is StandaloneDeploymentProfile. Browse for the configuration data file and select which host to deploy it to. The selected host must be the same as the SAML router is runing on. Click Finish.

___14 After the deployment is done, the SAML Session Provider will start and be in status "Starting" for up to 2 minutes. The reason for this is that the actual IFS configuration takes place the first time the SAML Session Provider is started and not during the deployment. When the IFS setup is finished, the SAML Session Provider should be put into status "OK". If there was no other session provider previously installed and activated, the SAML Session Provider is now the active session provider.

❑ Configure Infor Federation Services for the SAML Session Provider

___1 Log in to the IFS/AD FS server and log on to the "Infor Federation Services" application as an IFSApplicationAdmin

___2 Select Configure > Applications and select the newly created application (at the bottom of the list).

___3 Link security roles (grid-admin, grid-poweruser, grid-user, grid-runas, TestRole, and any other required roles) to the new application and save.

___4 For testing, add the grid-admin security role to the active account.

❑ Add Assertion Consumer Service endpoint to AD FS

___1 Find the federation metadata URL for the SAML Session Provider:

a From the Grid Management Pages, open the management pages of the SAMLSessionProvider application.

b Select Metadata.

c Copy the federation metadata URL displayed on the page for use in step 6.

___2 Log on to the IFS/AD FS server, and start "AD FS 2.0 Management."

___3 Expand "Trust Relationships" in the left side menu and select "Relying Party Trusts."

___4 Select the application that corresponds to your SAML Session Provider installation.

___5 Right-click and select Properties.

___6 On the Monitoring tab, enter the federation metadata URL for your SAML Session Provider (see step 1c for the value).

___7 Click Test URL to make sure that the address is reachable and trusted by AD FS. If you get an error message, see the Microsoft Windows Server documentation on troubleshooting trust management problems with AD FS 2.0.

___8 When you get a message saying that the URL was validated successfully, click OK and then OK again.

___9 Select again the application that corresponds to your SAML Session Provider installation.

___10 Right-click and select "Update from Federation Metadata."

___11 On the Endpoints tab, verify the SAML Assertion Consumer Endpoints, and then select Update.

❑ Test the SAML Session Provider installation

___1 In LifeCycle Manager, right-click on the Grid and select General tasks > URL > Java web start.

___2 After the Grid Management tool is started, ensure that the SAML router and the SAML Session Provider are started.

___3 Click on [login] in the bottom left corner. Provide the credentials used for IFS admin (which was given the grid-admin role).

___4 When the login succeeds, the "<not logged in>" is changed to the user name of the logged in user. Hover the cursor over the user name to see the provided roles. Ensure that the grid-admin role is assigned to the user.

___5 Configure the test servlet. See "Configure the test servlet" on page 53.

❑ Configure the test servlet

For test purposes, the saml-session-provider-gar contains a web servlet, com.infor.gridextension.sessionprovider.webapp.HelloServlet. When the IdP has been configured, this servlet can be set up to test the communication between the grid and the IdP. Setting this up also helps with understanding how roles should be configured in IFS and the grid.

___1 If you want, you can try to access the test servlet at this point to see what happens when a role hasn't been configured. The servlet should be available at **`https://[`***`SAML router IP address`***`]:[`***`SAML router https port`***`]/test/hello`**. The expected behavior is to get an error message saying that you don't have the required role for this application.

___2 After trying this, you will have a grid session set for your user. In order to make further tests after setting the roles, remove this session. Click Advanced on the Grid Management start page, then Sessions, and then remove the desired session. On this page, you can also see which roles have been mapped for your user.

___3 Create and map roles for the test.

The test servlet has been set up for role-based access control in its web.xml file. If you examine the web.xml file, you will see that only users with the role TestRole are allowed to perform a GET operation on the servlet. To give a user the TestRole role, you must create that role in IFS and then map the role in the grid.

a Make sure that you have configured IFS and AD FS to emit Security Roles as claims.

b Create a role called TestRole in IFS, according to "Add security roles in IFS" on page 53. Make sure to give the role to your test user.

c The role must be explicitly mapped in the grid. On the Grid Management start page, click Applications, then Configuration for the SAML Session Provider, and then Edit Role Mappings.

You will see the TestRole in the list of available roles. This is the role name that is referenced in the web.xml file.

Then click Edit... in the Included Members column of the TestRole, and click Add.... In the Global box, add TestRole as a custom role name. This is the role name that arrives in the claim from IFS. Remember to save your changes.

d Access the test servlet as described in the first step. The expected behavior is to get the message "Test successful."

Note: The roles in IFS and the grid could very well have different names, but this is not a problem as long as they are mapped correctly in the grid.

❑ Add security roles in IFS

___1 Start the Infor Federation Services application and log on with a user that is an IFSApplicationAdmin.

___2 Select Manage > Master Data.

___3 Double-Click on Security Role.

___4 Click on the New button and add the name of the Security Role (called Node name in the UI) and a description.

___5 You may assign users now to the new role by clicking the Add User button.

___6 Click Save when finished.

___7 Select Configure > Applications in the menu.

___8 Select the application that should be emitting this role

___9 Select all roles that the application should emit and click Save.

___10 To add additional users to the role after it is created, either:

- Enter the Manage > Master Data and add users to a role.
- Enter the Manage > Users, select a user, and add the role to that user.

The roles should not be emitted as Claims in the SAML Assertion token.

Configuring Assertion Consumer Services

In order to authenticate a given user, the SAML Session Provider sends an authentication request to the identity provider (AD FS 2.0). The response (assertion) is returned to one of a set of pre-configured assertion consumer service locations. These are endpoints where the SAML Session Provider receives and handles assertions from AD FS.

When a web application in the grid requires a session, this session is set as a cookie on the HTTP response. It is important that the assertion from AD FS is sent to the same host address as the one used in the original request from the client. Otherwise, the session will be set on the wrong context, and the client will not be able to access the desired resources. Both the SAML Session Provider and AD FS 2.0 must be configured to use the correct assertion consumer services.

If you access secured web applications in the grid via a proxy, you must add assertion consumer services representing the proxy host.

Initial Configuration

By default, the LifeCycle Manager installer will set up two assertion consumer service endpoints in the configuration: one for the FQDN, and one for the IP number of the SAML router. Unfortunately, only one of these can be automatically set up in AD FS. The installation must therefore be completed with a manual step, as described in the installation procedure for the SAML Session Provider. See "Installing and Configuring the SAML Session Provider" on page 47.

Updating Assertion Consumer Services

This section describes how to add more assertion consumer services to the configuration of the SAML Session Provider and AD FS 2.0. This is needed if you access secured web applications running in the grid via a host and port other than those already specified as assertion consumer service endpoints, for example, via a router different from the SAML router, or if you have a proxy in front of the grid.

Before you start This procedure assumes that the initial configuration of assertion consumer service endpoints has already been performed.

To update assertion consumer services

1 Add an assertion consumer service endpoint to the SAML Session Provider:

   a From the Grid Management Pages, open the Management Pages of the SAMLSessionProvider application.

   b Select Assertion Consumer Services.

   c Type the desired host address and port, and then select Generate ACS URL.

   d Click on the disk button to save your changes.

2 Add the assertion consumer service endpoint to AD FS 2.0:

   a Log on to the IFS/AD FS server, and start "AD FS 2.0 Management."

   b Expand Trust Relationships in the left side menu and select Relying Party Trusts.

   c Select the application that corresponds to your SAML Session Provider installation. Right-click and select "Update from Federation Metadata."

   d On the Endpoints tab, verify the SAML Assertion Consumer Endpoints, and then select Update.

Uninstalling a SAML Session Provider

To uninstall a SAML Session Provider, see the general instructions for uninstalling applications in the *Infor ION Grid Administration Guide for LifeCycle Manager 10*. In addition, note the following:

- When you uninstall a SAML Session Provider, the SAML router created during installation does not get uninstalled. It can be re-used for a new installation, or removed manually.
- The IFS/AD FS configuration does not get automatically removed. The application in IFS and Relying Party Trust in AD FS should be manually removed.
- Similar to the case of uninstalling a SAML Session Provider, if the installation of a SAML Session Provider fails, the IFS and AD FS configuration may need to be manually removed.

Error Handing for the SAML Session Provider

Active clients (non-browser) use SOAP calls using the WS-Trust standard for authenticating connecting users. In case anything has been set up wrong, different error messages may be provided by the Identitiy Provider (IdP).

HTTP Status	Description
200 (OK)	A successful login contains a RequestSecurityTokenResponse signed by the IdP.
500 (Internal Server Error)	SOAP Faults returned. See descriptions below.
503 (Service Unavailable)	The 503 status is returned if the service is not available on the server. That can mean that the WS-Trust usernamemixed service is deactivated in AD FS. Indicates configuration issues. Ensure that in AD FS the Endpoint "/adfs/services/trust/13/usernamemixed" for WS-Trust 1.3 is both Enabled and Proxy Enabled.
502 (Connection Failed) 502 (DNS Lookup Failed)	The 502 status can mean that the server does not listen to that port (Connection Failed) or that the server could not be found (DNS Lookup Failed). Indicates configuration issues.

SOAP Fault	Description
ID3242: The security token could not be authenticated or authorized	Logon failed. Either the username does not exist or the password was wrong or the user dows not have access to the application.
ID3082: The request scope is not valid or is unsupported.	This response is returned if the service provider RelyingPartyTrustIdentifier defined in the send SOAP message does not exist or is configured wrong. This can mean that the setup failed and that AD FS and IFS are not correctly configured. It may also mean that the Session Provider Entity ID is wrong in the SAMLSessionProvider properties in the grid.
MSIS3127: The specified request failed.	This response is returned if the ADFS could not understand the XML request part of the SOAP message.

Changing the Session Provider

The grid comes with a built-in session provider, the developer session provider, that accepts all users and passwords. Do not use this session provider in a production scenario!

To define a session provider

1. Install an application that can act as a session provider, that is, provides services according to the SessionProvider interface hierarchy.
2. Grant session provider rights to the selected application from the grid Configuration Manager.

   a. On the Configuration Manager page, click the Session Providers link.

      The Session Providers page will appear. It displays information about the registered session provider and contains a drop-down list of eligible applications. Of these applications, two are pre-populated entries: "No Session Provider" and "The Developer Session Provider, intrinsic to the grid."

   b. Select the preferred application and click the Grant button.

   c. Save the configuration changes by clicking the diskette icon at the top left corner of the page.

      Note that the changes are applied immediately, and the Session Providers page is updated to reflect the new configuration.

   Note: There might be a perceived discrepancy between the Registered Session Provider field and the Application granted Session Provider rights field on the page. The latter can contain more entries than the first. This is because when you register a session provider, you are actually registering an application type (the name in between the parentheses in the drop-down list), not a deployed application instance. You will notice this behavior if you have an application type that is deployed multiple times under different names. The result is that all the instances will be granted session provider rights as indicated under the Application granted Session Provider rights field.

Configuring Router WWW Authentication Methods

Each router in a grid can be configured to provide specific authentication methods for its HTTP and HTTPS ports, respectively. The reason for this is to allow different authentication methods via different entry points in the grid. It can also be desirable to force use of basic authentication, for example, to go over the HTTPS port.

The WWW authentication methods configured for a router define the methods accepted by the router when accessed via the HTTP or HTTPS ports. When accessed by a grid client, the intersection of these methods and the methods supported by the configured session provider determines how authentication is performed in practice. For information on the supported authentication methods for

each session provider, see "Session Provider Requirements and Selection" on page 27. Basic authentication is supported by all session providers.

If SAML 2 authentication is configured for a router, it will override the other settings. The SAML Session Provider must be the configured session provider in order for this setting to take effect. Note that when the SAML Session Provider is installed, a router (SAML Router) configured to use SAML 2 authentication is automatically installed. The SAML 2 authentication method in the SAML Session Provider uses WS-Federation to authenticate to AD FS 2.0 (for browser clients that can be automatically redirected). The SAML Session Provider also implements basic authentication using WS-Trust to authenticate users to AD FS 2.0 (for active, non-browser based clients).

It is possible that the configured session provider does not support any of the configured WWW authentication methods. In this case, the user will not be able to log on.

To configure WWW authentication methods for a router

1. Navigate to the Configuration Manager for the grid you are working with.
2. Click Routers.
3. Click on a router and select the WWW Authentication Methods tab for the HTTP or HTTPS port.
4. Select the appropriate authentication methods and click Update. (Note that the router will automatically restart if these fields are modified.)
5. Click Save and then Save again.

Authenticating with a Grid Client Certificate

When SSL is used to connect to the grid, it is possible to log on users with a grid client certificate. Note that no session provider is involved when the grid principal and session are created when the SSL handshake authentication mechanism is used. The roles given to the user are those specified in the client certificate.

Authenticating with a Grid Client Certificate to the Grid Management Pages

As an example of connecting with a grid client certificate, we will describe how to use this method when accessing the HTML-based Grid Management Pages over HTTPS. This is particularly useful when grid-admin access is needed and there are no users with the grid-admin role set up via the role mappings page.

For more information about the HTML-based Grid Management Pages, see the *Infor ION Grid Administration Guide for LifeCycle Manager 10*.

To connect to the HTML-based Grid Management Pages over HTTPS with client authentication

1 Create a grid client keypair and certificate with the grid-admin role, by using the console command **`create=clientcert`**. Use the options **`–role grid-admin`** and **`–keystoretype=pkcs12`**, along with the other required and desired options. For complete syntax, see "Console Tool Guide" on page 12.

   A keystore with the extension .p12 will be created in the client keystore directory. Remember the password entered to generate the keystore.

2 Import the generated client keystore into the appropriate location. This will differ depending on the system and browser you are using. For Windows and Internet Explorer or Chrome, do the following:

   a Double-click the .p12 keystore to open the Certificate Import Wizard.

   b When asked to enter the password for the private key, enter the password given in step 1.

3 Configure a grid router for HTTPS with client authentication:

   a In the Configuration Manager, select Communication > Routers and then select the router to configure.

   b In the HTTPS configuration area, make sure that the Port field has a value.

   c In the HTTPS configuration area, make sure that the HTTPS Authentication Type is set to either "Clients may authenticate with certificate" or "Clients must authenticate with certificate."

   d If the configuration was changed, restart the router.

4 With the appropriate browser, navigate to the following URL: **`https://`***`server`***`:`***`port`***`/grid/ui`**

   where *server* is the name of the server hosting the grid router configured in step 2, and *port* is the HTTPS port for that router. A grid session will be created, based on the information in the client certificate.

Authorization

4

Authorization Overview

The access control in the grid and in grid applications relies on access policies. The access control process can be divided into two steps:

1. Policy definition, where each method in each grid proxy and web service in the grid and grid applications may define the authorization level required to use each individual function.
2. Policy enforcement, where access requests are approved or rejected based on the required authorization level of the called method and the assigned roles of the caller.

Authorization is the function where access requests are approved or rejected based on the previously defined authorization levels.

See the documentation for each grid application for details on what roles there are for that specific application, and what the roles are used for.

Authorization Levels

There are five levels of authorization in the grid and its applications: Public, Restricted, Private, Session-based and Session-based with role restrictions. The authorization mechanism is in effect regardless of the type of client.

Public

The Public authorization level is for methods and functions that require no authorization. It is not even necessary to be authenticated to the grid to be able to run these functions. It is similar to public web pages on the Internet. This is the default authorization level unless the grid application specifies any other.

Restricted

Methods that are set to the Restricted authorization level only provide access to grid applications, not to users or external clients.

Private

Methods that are set to the Private authorization level only provide access to the grid application that owns the method. No other grid application, users, or external clients may use the methods.

Session-based

This authorization level is for methods that require the caller to be a valid authenticated grid user, but it is not necessary that the user belongs to any specific role or group. Any caller with an active valid grid session (or a caller having authenticated with a certificate) may use the function.

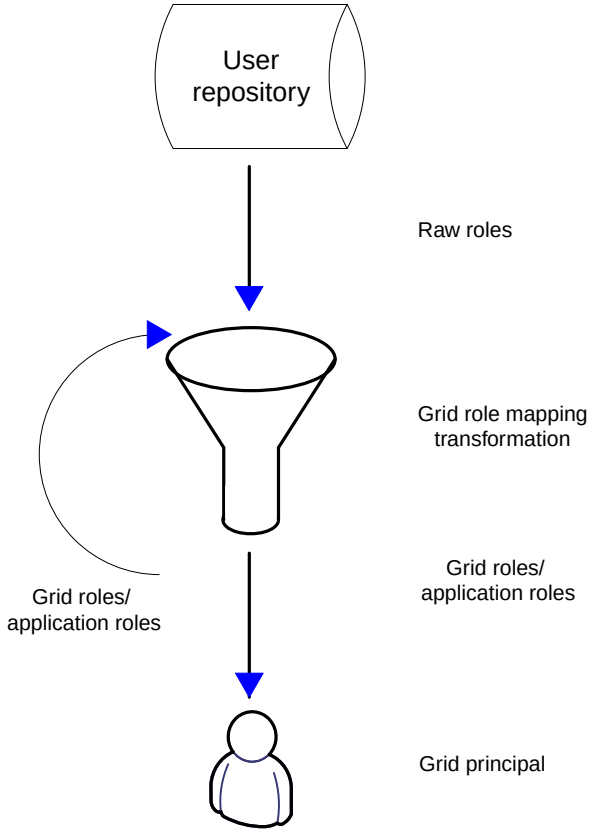
Session-based with Role Restrictions

This authorization level is for methods that require the caller to have a specific role, in addition to being authenticated. Each function specifies the roles that are required to be able to access the function. For information on how to assign users/groups to grid application roles, see "Defining Role Mappings" on page 64.

How Roles Are Assigned to Users

When discussing roles, we distinguish between the raw roles that emanate from the user repository, and the grid roles that are the result of applying the grid role mapping transformation to the raw roles.

Figure 2. Illustration: Role assignment process



During authentication, the active session provider retrieves raw role information from the user repository. The username is also considered a raw role. The grid role mapping transformation is applied to the raw roles, and the resulting grid or grid application roles are assigned to the grid principal. A grid role may be transformed any number of times to additional grid roles. In other words, once role A has been assigned to a principal, having role A may lead to role B also being assigned, if the role mapping has been set up that way.

At runtime, the grid and its applications can query the grid principal for role membership to make authorization decisions.

For more information about grid principals, see "Grid Principals and Sessions" on page 25. For details on setting up role mappings, see "Defining Role Mappings" on page 64.

Certificate-based Authentication and Roles

Clients connecting to the grid with a certificate may include roles when the certificate is created. See "Console Tool Guide" on page 12. The roles in the certificate are treated as raw roles in the same way as when a session provider is used.

Session Providers and Roles

Each session provider retrieves raw role information in a different manner, as described in the following sections.

LDAP Session Provider

The LDAP Session Provider queries the configured LDAP server for the group memberships of the authenticated user. Even though there may be several LDAP servers configured, each user ID may only be present in one LDAP server. If a logon attempt finds the provided user id in more than one LDAP server, an exception will be thrown and the login will fail. The groups (raw roles) of the user are gathered from the same LDAP server as the user was found in.

When the grid role mapping queries the LDAP Session Provider for available groups, the session provider makes calls to all the configured LDAP servers to get a list of all available groups.

Windows Session Provider

The Active Directory is used to get the group membership (raw roles) for the authenticated user, as well as the available groups for the session provider roles for the grid role mapping.

DSSO Session Provider

The DSSO Session Provider does an extra call to the Lawson Security system to get the role memberships of the authenticated user. The same mechanism is used to get a list of all available roles in the Lawson Security system for the session provider roles for the grid role mapping.

SAML Session Provider

The SAML Session Provider receives the authenticated user's raw roles in the Security Role claim from the Security Token Service (AD FS). The Security Roles are created, configured and assigned (both to application and to users) in the IFS management tool.

When the grid role mapping queries the SAML Session Provider for available groups, the session provider makes calls to the IFS web services to get a list of all available Security Roles in IFS. Not all the Security Roles in the list from IFS are necessarily emitted by the grid application in IFS. Compare with the Security Roles assigned to the grid application in IFS for details, since the output from the IFS web service can not be filtered per application.

The SAML Session Provider has two application properties that are essential to be able to query for groups (raw roles). Those are the "IFS administrator" and the "IFS administrator password". They are required to be able to connect to the IFS Web Services. For details on required account permissions, see "Installing and Configuring the SAML Session Provider" on page 47.

Global Roles and Application Roles

Roles are used in the grid and in grid applications to provide access control for applications in general or for specific functions in an application. A global role is valid in the entire grid whereas an application role only can be referred to from the application itself.

Default Roles

There are three default role levels for the grid and for all applications: administrator, power-user and user. These roles each have a different level of access.

Administrator

There are administrator roles for the grid itself and for the applications. The roles are named "grid-admin" for the grid and "<Application name>/app-admin" for each application.

The grid-admin role grants full access to the grid internal functions. It can be used to install/uninstall applications, change all properties, add/remove hosts, and so on. This role should be assigned to users with great care.

The app-admin role grants full access to administer the application in the grid management user interface. It does not necessarily give access to using the application itself.

Power-user

The power-user role in grid is called "grid-poweruser" and for applications "<Application name>/app-poweruser".

This role grants access to a limited set of operational tasks, for example, log level settings and application level operations

User

The user role in the grid is called "grid-user" and for applications "<Application name>/app-user".

The "grid-user" role is currently not in use. The application user roles may be used by the application. Review the documentation for each application to see how roles are used in that application.

Application-specific Roles

Each application may define additional roles to use for authorization purposes. Review the documentation for each application to see which roles are used in that application.

Defining Role Mappings

Use this procedure to configure role mappings for a grid.

The grid role mapping is a way to transform existing groups, roles or usernames from the user repository into grid roles and grid application roles. Each session provider uses different methods to retrieve groups and roles (see "How Roles Are Assigned to Users" on page 61).

You can configure role mappings belonging to a particular application on a configuration manager page belonging to that application (a page with an application centric focus) or you may configure role mappings on a global page enabling you to operate on all application defined roles including the ones defined by the ION Grid itself (grid-admin, grid-poweruser & grid-user). Which is best for you depends on the situation.

Navigation to Role Mapping Pages

How you navigate to the pages for editing role mapping depends on whether you want to work with the global role mappings page (where the roles for all applications are available) or with an application-centric role mappings page (where only the roles for one application are available).

To navigate to the global role mapping page

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click the Users and Role Mapping link.

   A page with all grid and grid application defined roles will be shown and can be configured.

To navigate to an application-centric role mapping page

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click the Applications link.
3. Select the application you would like to configure role mappings for.
4. Click the Edit Role Mappings link at the top of the page.

   A page with all available grid application-defined roles for this application will be shown and can be configured.

Regardless of whether you used the global or the application-centric role mapping page, you will now be presented with a page that lets you define and edit role mappings.

Configuring Role Mappings

When you configure role mappings, you will need to select a role and the type of mapping (inclusive or exclusive) that you want to use. You perform this selection on the first role mapping page, which has the following columns:

Column	Description
Roles	The name of the role.
Included Members	A list of included users/groups/roles that should be mapped into the grid role. Select the "Edit…" link to add or remove from this list.
Excluded Members	A list of excluded users/groups/roles that should be mapped into the grid role. Select the "Edit…" link to add or remove from this list.
Description	A description of the grid role. Hover over the string if it is too long to display in full.

The included/excluded lists can be used to grant access to a subset of users in a group. Assume that group Alpha is granted access to grid-admin, and that group Beta is excluded from it. Any member of both Alpha and Beta will NOT get the grid-admin role. Only users that are members of Alpha but NOT Beta will be grid-admins. The included/excluded lists can contain both role names and individual users.

To add a group to the Included Members list

1 After navigating to the role mappings pages as described above, identify an application-defined role that you want to configure mappings for from the list.

2 Click the Edit link in the Included Members column. It is important to note that once you click the Edit link for a particular role, you are in the context of that role. Any mappings will be for that role. For example, if you clicked Edit for the grid-admin role, all mappings you configure will be for that role until you return to the list of roles and select a different role.

   After you click Edit, a new window opens named "Role mappings" with the text "Members included in <*roleName*>". On the left side there is a list of included members or the text "<no included members>" if no mapping has been configured yet.

3 Click Add... to add role mappings.

   After you click Add..., the Add Role Mappings window appears. In this window, you will make new mappings for the selected role, for example, for grid-admin. There are three different sources to map groups/roles into Grid roles from. See "Selecting Role Mappings" on page 66.

4 Select the source whose roles you want to map to the role you selected above, and then click Add on the same row where the selection was made. It is not enough to click OK at the bottom. Every added group must be added using the correct Add button before adding another mapping. When done adding members, click Ok to confirm.

5 In the Role mapping window, the added members should appear in the last on the left. Click OK to confirm.

6 In the window with the list of roles, click Save in the upper left corner.

Selecting Role Mappings

There are three different sources of roles/groups and users for the mapping: the global source, the application source and the session provider source.

Global

The Global source is for defined global grid roles or for custom written mappings.

The defined list includes the default grid roles: grid-admin, grid-poweruser and grid-user. This list also includes the "authenticated" role, which includes anyone who has authenticated to the grid and has a valid active grid session.

The Custom field is a free text field where a user id or raw role can be added. Mappings to these free text items requires an exact match

Application

The Application source is for defined application roles or for custom application roles.

The defined application roles are a list of all the available roles for all grid applications used in the grid. For example, by doing this you can say that anyone who is an app-admin of application A can also be an app-admin of application B.

The Custom application mapping is done by selecting an application in the drop-down box and then entering a role in the free text field.

Session Provider

The Session Provider source uses the active session provider to get all the defined groups/roles in the user repository. The Defined list contains all the available groups/roles.

It is possible to filter the search using the Filter free text field and clicking the Update button.

Password Management

Grid application passwords are stored encrypted in the database with the rest of the grid and application properties. Encryption and decryption of the password property is made with a grid-unique symmetric key and is handled by the property management layer in the grid.

The grid provides functionality for exporting and importing application properties, which can be useful, for example, for re-using configuration. When exporting application properties that include passwords, a password to protect the password properties must be given. When importing back the properties, the same password must be entered again in order to decrypt the information.

Logging and Auditing

5

The grid framework provides extensive logging capabilities. Each running grid node has its own log file.

Logging Levels

There are six different logging levels that can be individually enabled. There is a seventh log level configuration option, the "All" option, which enables all logging levels.

The logging level configuration is inheritative. There is a master log level setting for the grid, accessed from the Logging > Log Levels page in the Grid Management Pages. If nothing else is configured, this log level will be the same for each application and each grid node. The grid log levels which are enabled by default are: Error, Warn, Note and Info.

It is possible for each application to define its own log level, which will override the log level of the grid. In a multiple-hosts grid, it is possible to set the log level per host per application.

When enabling the Debug or Trace log levels, it is recommended to do so only for the application to debug and not for the entire grid, due to the amount of log entries generated.

The available logging levels are described in the following sections, in order of decreasing criticality.

Error

Error is reserved for special exceptions/conditions where it is imperative that you can quickly pick out these events. It is intended to be used for error messages which indicate a major problem which must be investigated and resolved and either has stopped or will stop operation of the application.

- An error has occurred in the program (usually an exception).
- Severe errors have caused premature termination.
- Other runtime errors or unexpected conditions have occurred.

It is not possible to turn off the Error log on a global basis.

Warn

An anomalous condition has been detected and the program will attempt to deal with it. This logging level indicates a problem which should be investigated and resolved, but does not seriously impact operation of the application. The Warn level is typically used under the following conditions:

- When a threshold level is reached.
- A loss of connectivity occurs that can be repaired by reconnecting.
- Use of deprecated APIs is detected.
- Poor use of APIs is detected.
- Undesirable or unexpected runtime situations occur, but they are not seriously wrong.

Note

The Note log level contains less critical information, which indicates a problem that does not necessarily need to be investigated but should be logged.

Info

The Info level is typically used to output information that is useful to the running and management of your system. It would also be the level used to log entry and exit points in key areas of your application.

Debug

The Debug logging level provides detailed information on the flow through the system used to identify possible errors or misconfiguration in the runtime environment. Debug-enabled logs may contain output information for various critical functions.

The Debug log level should not be activated in normal program operation, since it fills up the log files quickly.

Submitted error reports should preferably contain logs which have the Debug logging level enabled for easier and faster resolution. Normally, if a case is escalated to the development organization, the first thing asked for is trace/debug logs that show the problem.

Trace

The Trace logging level represents the highest level of detailed information on the flow through the system. This level can be used to identify where the call goes, which methods that are involved and output.

The Trace logging level should not be activated in normal program operation, since it fills up the log files quickly.

Submitted error report should preferably contain logs which have Trace enabled for easier and faster resolution. Normally, if a case is escalated to the development organization, the first thing asked for is trace/debug logs that show the problem.

Configuring Logging Levels

The log level can be configured on multiple levels. The configuration is found in slightly different ways for each level.

Grid-wide Logging Levels

To configure grid-wide logging levels

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click Grid Properties.
3. Find the Node log level property and click on the link in the Value column. The link is either "<undefined>" or a list of log levels such as "ERROR,WARN,INFO,NOTE."
4. Select which log levels to have. Click Save.
5. Click the Save button on the upper left corner.

Router Logging Levels

To configure router logging levels

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click Grid Properties.
3. Find the Node log level property and click the "Node log level" link (not the link in the Value column).
4. To change the log level of a particular router, click on the link in either the "Any host" column or the column for a particular host. If a particular host is selected, the logging levels set will only affect the selected router on that host.
5. Click the Save button on the upper left corner.

Application Logging Levels

To set the application log level it is either possible to follow the instructions for router levels above or to do the following:

To configure application logging levels

1. Access the Configuration Manager as a user with the grid-admin role.
2. Click Applications.

3 Click on the application to configure log levels for.

4 Click on Edit Properties.

5 Find the Node log level property and click on the link in the Value column. The link is either "<undefined>" or a list of log levels such as "ERROR,WARN,INFO,NOTE."

6 Select which log levels to have. Click Save.

7 Click the Save button on the upper left corner.

Application-level Logging for a Specific Host

To configure application-level logging for a specific host

1 Access the Configuration Manager as a user with the grid-admin role.

2 Click Applications.

3 Click on the application to configure log levels for.

4 Click on Edit Properties.

5 Find the Node log level property and click the "Node log level" link (not the link in the Value column).

6 On the Property:Node log level page it is possible to set the log level for all hosts, as well as for individual hosts. To change the log level, click on the link in either the "Any host" column or the column for a particular host. If a particular host is selected, the logging levels set will only affect that host.

7 Select which log levels to have. Click Save.

8 Click the Save button on the upper left corner.

ION Grid Installation Scenarios

6

Recommended ION Grid Installation Scenarios

It is recommended that SSL is used to encrypt the traffic between client and server whenever the data operated on is of a sensitive nature. This reduces the risk of successful network sniffing and similar types of intrusions. When SSL encryption is used, following the initial SSL handshake between the client and server all traffic is encrypted using the specified cipher.

SSL is used for internal grid communications between hosts and applications.

It is possible to install the Grid in various different scenarios.

Cloud

When the grid is deployed in the cloud, all SSL certificates should be signed by a Public Certificate Authority to enable automatic trust in browsers and other devices.

To reduce risk to the confidentiality and integrity, only the HTTPS ports should be used on the Grid routers.

To further reduce the risk of intrusions it is suggested to use some web filtering mechanism and intrusion detection/prevention system.

User repositories must be protected and not directly accessible from the Internet. It is recommended to require some kind of VPN connection for user management.

Internal

As mentioned, it is always more secure to use SSL communication between clients and the Grid routers. This will reduce the risk of internal intrusions such as network sniffing and data loss.

Internal Supporting External Users

The scenario of having an on-premise installation serving internal users that also provides some access to external users/devices can be tricky. It is NOT recommended to publish grid routers directly accessible

from Internet without having some filtering proxy in between. The recommended way to enable access to external users/devices is to have a VPN setup in a DMZ that users connect to.

Open communication on the internet (that is, any communication which takes place with clients outside the secure intranet) always carries an additional risk in terms of enabling the potential for security breaches.

Here we will discuss two options for allowing grid router access over the Internet to a grid application running on the intranet.

VPN

Infor strongly recommends that any client wishing to access a grid application from the Internet utilize a VPN tunnel to provide a secure connection between the client and server. There are a number of reasons why this method is preferable:

- Secure - VPN offers a highly secured solution with a reduced ability for intrusion attacks to take place.
- No client installation – Internet access and a web browser are the only requirements to create the tunnel to the corporate network without user or administrator overheads.
- Support for any client-server architecture which can be accessed over the normal corporate LAN.
- Ability to configure and restrict user access. For example, a contractor can be limited to only be able to connect to a specific grid router, so this does not fully open up the corporate network unless you want it to.
- User access control is made simpler since there is one entry point for the corporate network . An administrator can configure and secure the system by allowing/denying users access to specific areas or the whole corporate network. this is a strong consideration when preventing someone accessing the system, for example, following termination.
- VPN solutions are available as physical or virtual appliances giving different options for deployment depending on existing infrastructure.

Direct Connect – Client connection through the DMZ using SSL

As an alternative to the use of a VPN solution, it is possible to implement an SSL-encrypted route through the DMZ for clients to connect to.

This involves configuring static port forwarding such that changes are made to the external and internal firewalls to allow routing of traffic from a specified external port to an internal LRC router.

Configuration involves the following high-level steps:

1. Configure the grid router, determine the port which is secured, and ensure that encryption is enabled.
2. Select and open a port in the external firewall, configuring a static port forward to a selected port on the internal firewall.
3. Open the port specified in the previous step in the internal firewall, and configure a static port forward to the specified grid router port.

Considerations and notes:

1. A non-standard port must be opened in the external firewall and this has a rule based route to a port within the intranet on a specific host.
2. There is a risk with this approach since there is a direct route from the Internet to the intranet.
3. The security depends on the grid router, the active session provider and the security of each web application available through the published router.
4. All traffic is SSL encrypted.

Reference

A

ION Grid Terminology

Throughout this guide various terms and abbreviations are used that may require explanation.

Term	Description
Asymmetric encryption	Asymmetric encryption means that different (mathematically related) keys are used for encryption and decryption. The public key can be shared freely, while the private key must be kept secret. Asymmetric encryption is more often called public key encryption.
Public key	The public key is used to encrypt data intended for the holder of the corresponding private key. Only the private key can decrypt information encrypted with the corresponding public key. A public key is typically distributed by sharing a certificate that vouches for its authenticity. The public key can also be used to validate the authenticity of data signed by the corresponding private key. Such signatures are the basis of trust in a certificate – if the relying party can verify the issuer's signature (by having access to the issuer's public key), the public key of the certificate can be trusted.
Private key	The private key must be properly protected to ensure that it is indeed kept private.
Key pair	A matching pair of a public key and a private key.
Symmetric encryption	In symmetric encryption, the same key is used for both encryption and decryption. It is therefore very important to protect the key from unauthorized entities.
Certificate	A certificate vouches for the authenticity of a public key. It is a signed statement from an issuer stating that the included public key belongs to the holder of the private key. The issuer is called a Certificate Authority. A certificate can be signed by the private key itself and is called self-signed certificate.

Term	Description
Certificate Authority	A certificate authority (CA) is an entity that issues and revokes certificates for various uses. Public Certificate Authorities such as VeriSign, EnTrust and Thawte are by default trusted by the most common browsers. Certificates issued by those entities are automatically trusted by the browser if they are still valid. The certificates of a non-public (or internal) Certificate Authority are not trusted by clients automatically. This can be remedied by having the issuing certificate of the Certificate Authority imported into the Trusted Root Certificate store.
Certificate Signing Request	When a new certificate is required, a keypair is generated on the host where it will be used. The Public Key is bundled together with additional information like key usage and user/host information. The result is called a Certificate Signing Request and is packaged according to the PKCS#10 standard. A Certificate Signing Request is submitted to a Certificate Authority for signing. The resulting issued certificate should be in a format following the PKCS#7 standard.
CA	See Certificate Authority.
CSR	See Certificate Signing Request.
Keystore	A keystore is a collection of certificates and/or key pairs. The keystore is normally stored in a file on a hard drive. The grid uses various keystores as described in "Grid Keystores" on page 10.
Authentication	Authentication is the process of receiving provided credentials and ensuring that the credentials are matching and correct.
Authorization	Authorization is the process of validating if an authenticated used has the permission to access a specific resource.
Certificate Revocation List	The Certificate Authority may provide a Certificate Revocation List (CRL) that keeps track of all revoked certificates. In order to make use of the CRL, issued certificates must include a link to where the CRL can be found
CRL	See Certificate Revocation List.
Revocation	Revocation is the process of a Certificate Authority to invalidate an issued certificate.
Self-signed certificate	A self-signed certificate means that the signer of the certificate is the holder of the public key itself. That means that there is not any trust for the certificate by default. Extra measures must be taken in order for that certificate to be trusted by browsers and other clients/devices.

Term	Description
Grid-signed certificate	A grid-signed certificate is a certificate issued by the grid CA. The grid CA's issuing certificate is not automatically trusted by browsers and other devices. If the grid root certificate (issuing certificate) would be imported into every browser's and device's trusted root certificates store, then the grid CA becomes a trusted CA.
CA-signed certificate	A CA-signed certificate means that the issuer of the certificate is a trusted entity (either an internal corporate CA or an external CA). Browsers automatically trust the certificate since the CA certificate that performs the Issuing of the certificates is in the browser's list of trusted root certificates.
PKCS	Public-Key Certificate Standards issued by RSA Security for various cryptographic uses.
LifeCycle Manager (LCM)	An application lifecycle management utility used by Infor M3 and in some cases Infor Lawson. The LifeCycle Manager can be used to install and manage the grid and grid applications.
Bootstrap	The Bootstrap is the name of a component used to install and manage grids without the use of LCM. The Bootstrap is used by Infor ION to manage grid instances.
SAML	The Security Assertion Markup Language is an XML-based open standard data format for exchanging authentication and authorization data between parties, in particular between an identity provider and a service provider. The grid has a SAML Session Provider which uses the WS-Federation and WS-Trust standards for authenticating users when Infor Federation Services (IFS) is used. See "Installing and Configuring the SAML Session Provider" on page 47 for further information.
Identity Provider	(IdP) Identity Provider (also known as Identity Assertion Provider) is an authentication module which verifies a Security token as an alternative to explicitly authenticating a user within a security realm.
Service Provider (SP)	An entity providing a service. The SAML Session Provider is a Service Provider when using IFS. The SP authenticates the users to the IdP.

Index

A

C

D

E

G

I

K

L

M

P

R

S

U

W