Configuring the SSL connection
between
IBM Tivoli Identity Manager and the LDAP server

TABLE OF CONTENTS

1 Introduction

This document provides instructions on how to set up the SSL connection between the Tivoli Identity Manager Server and the LDAP directory server.

The instructions assume that you are creating and using self-signed certificates using IBM Global Security Kit (GSKit). If you use certificates issued from well-known certificate authorities (CAs), such as VeriSign, the instructions are slightly different from what is described in this document.

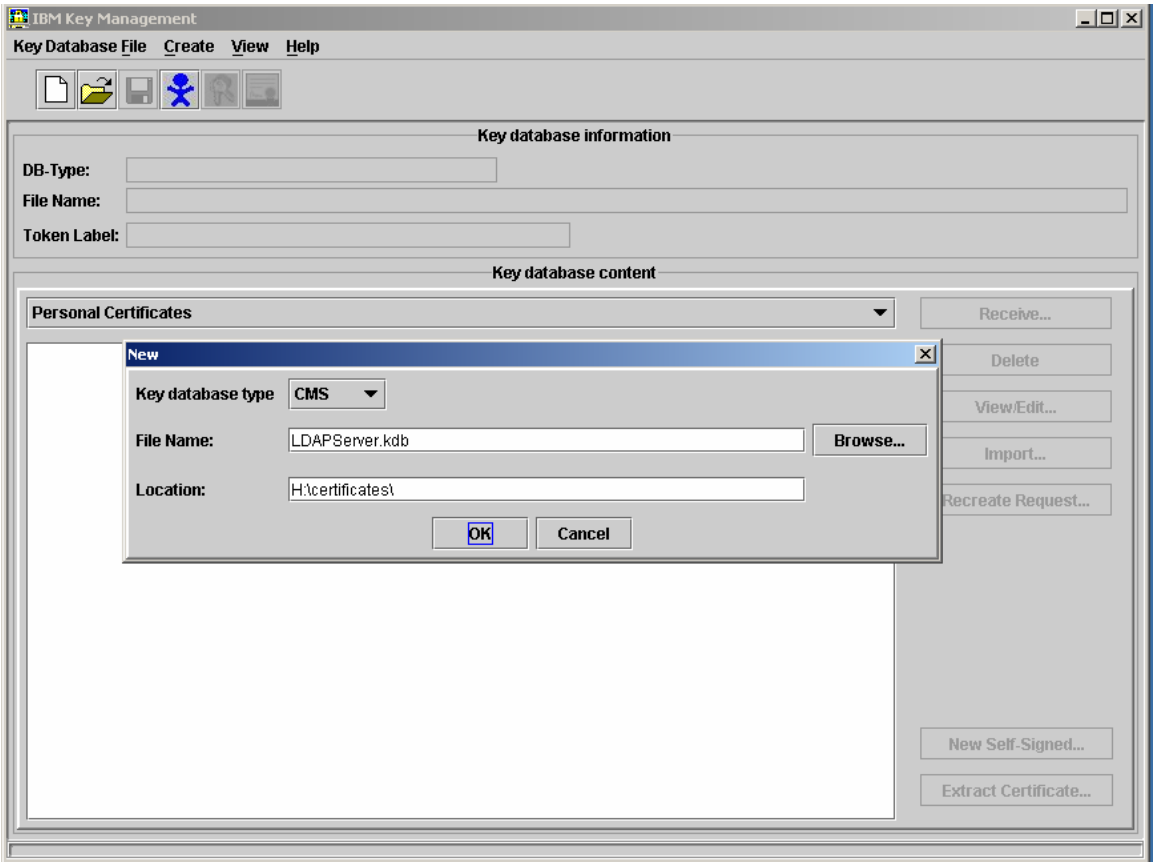
The instructions are based on using IBM Tivoli Directory Server, Version 6.0 and IBM Tivoli Identity Manager, Version 4.5.1 or Version 4.6, running in IBM WebSphere Application Server, Version 5.x.

2 Creating Self Signed Certificates using IBM GSKit

First, you need to create the self-signed certificate and the CMS key database using IBM GSKit, and its ikeyman utility. The self-signed certificate will be used by LDAP.

1. Start the ikeyman utility (ikeyman.bat or ikeyman.sh). It is located in the <WAS_HOME>/AppServer/bin directory.

2. From the **Key Database File** menu, select **New**.



3. From the **Key database type** drop-down list, select **CMS**.

4. In the **File Name** field, type the appropriate file name for the key database file. For example, type `LDAPserver.kdb`.

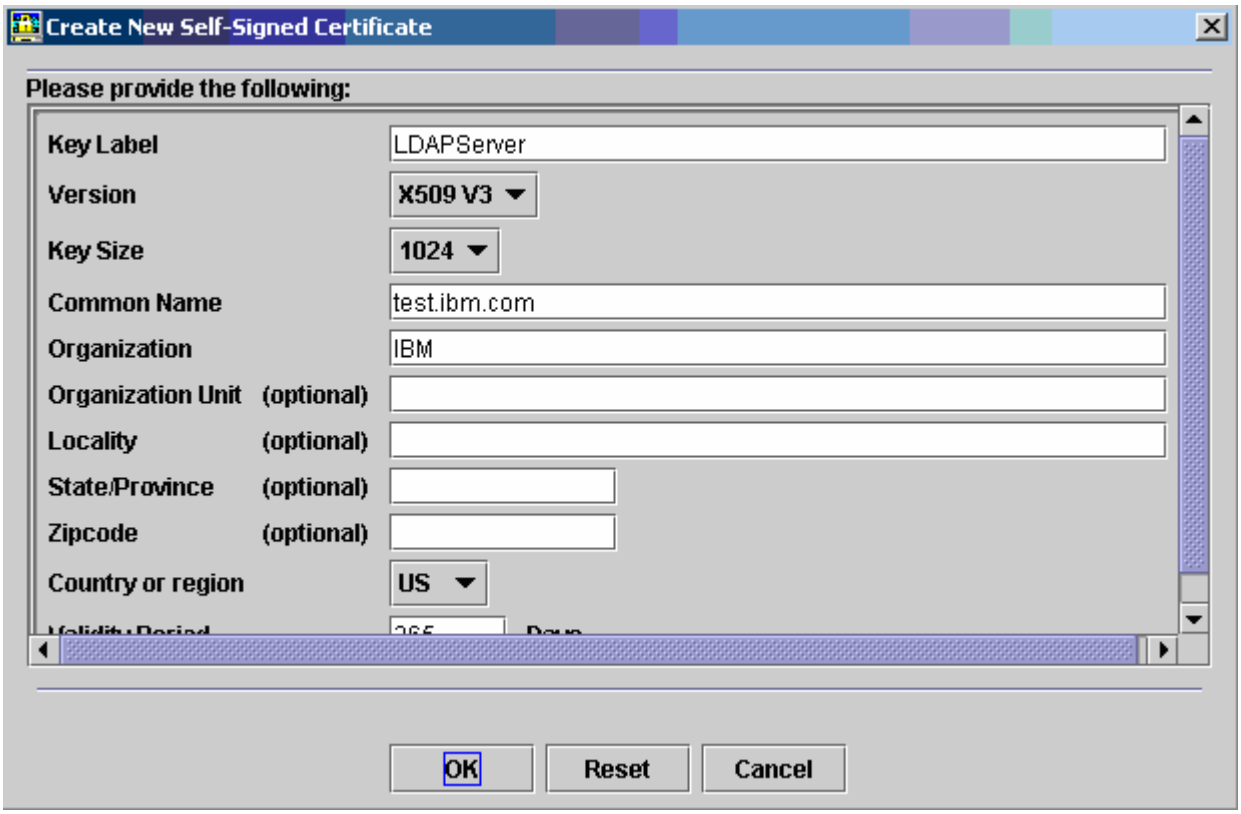
5. In the **Location** field, type the location where you want to store the key database file. For example, type H:\certificates\.

6. Click **OK**.

7. In the Password Prompt window, type a password (for example, use “secret” as your password) in the **Password** and **Confirm password** fields. Do not select Set expiration time. Select **Stash the password to a file** to save the password for the key database file in a file.

Next, you need to create the self-signed certificate for the LDAP Server.

1. From the **Create** menu, select **New Self Signed certificate**.



2. In the **Key Label** field, type LDAPserver, since you are creating this self-signed certificate for the LDAP Server.

3. In the **Common Name** field, type the machine name for the LDAP Server.

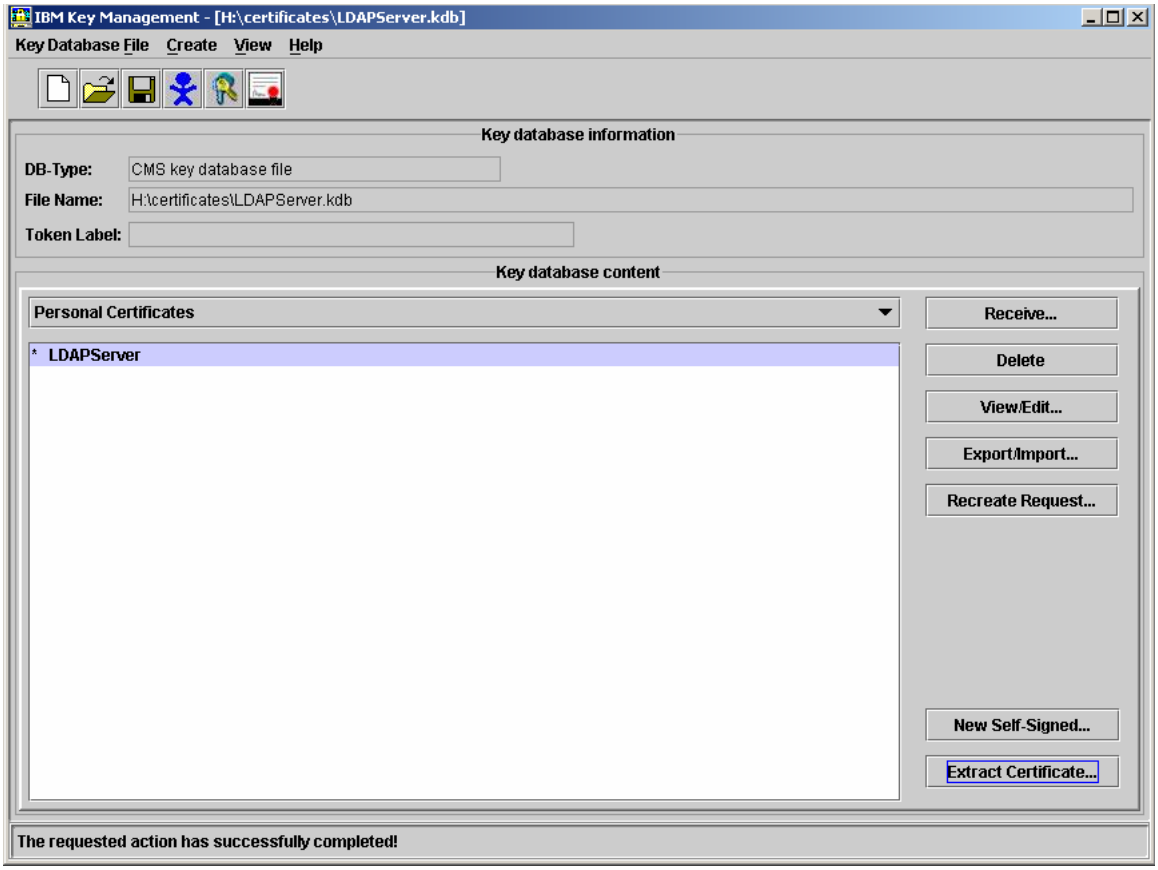
4. In the **Organization** field, type the name of your organization. For example, type IBM.

5. In the **Valid Period** field, type a value, if you want it to be longer than the default of 365 days.

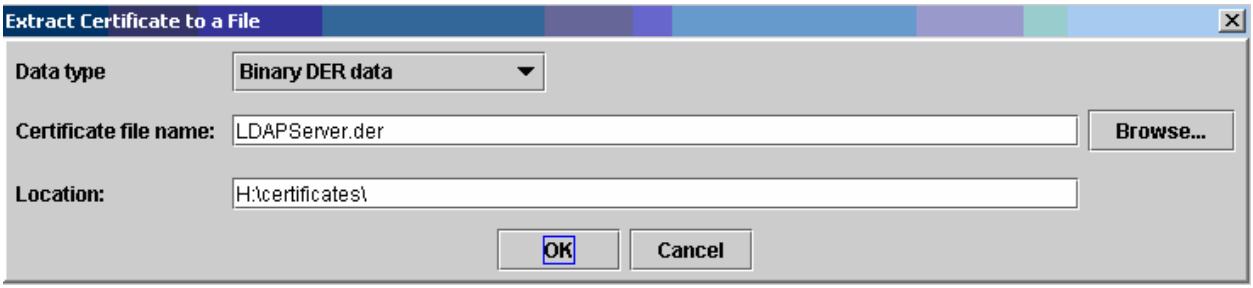
6. Click **OK** to create the self-signed certificate.

Next, you need to extract the certificate. This certificate, which will be a file with .der extension, will be used by the SSL client (IBM Tivoli Identity Manager Server in our case) while initiating a communication with LDAP server. The LDAP server will verify this certificate prior to accepting the connection.

1. On the main window, click **Extract**. A window similar to the following is displayed:



2. In this window, click **Extract Certificate**.



3. In the Extract Certificate to a File window, from the **Data type** drop-down list, select **Binary DER data**.

4. In the **Certificate file name** field, type the name of the certificate file. For example, type `LDAPserver.der`.

5. In the **Location** field, specify the location where you want to store this certificate file.

6. Click **OK**. The certificate is created in the file and location that you specified.

Finally, now that you have the self-signed certificate, you need to copy it to the server where Tivoli Identity Manager is installed. Copy the certificate file, for example the LDAPserver.der file, to a \certificates\ directory on the Tivoli Identity Manager Server.

3 Configuring SSL for the directory server

To have SSL communication between the LDAP server and Tivoli Identity Manager, the LDAP server must be configured to use SSL for secure communications. If you are using IBM Tivoli Directory Server or Sun ONE Directory Server to store Tivoli Identity Manager information, you must set the server to use SSL, and then configure the SSL certificates that you want to use.

3.1 Configuring SSL for IBM Tivoli Directory Server

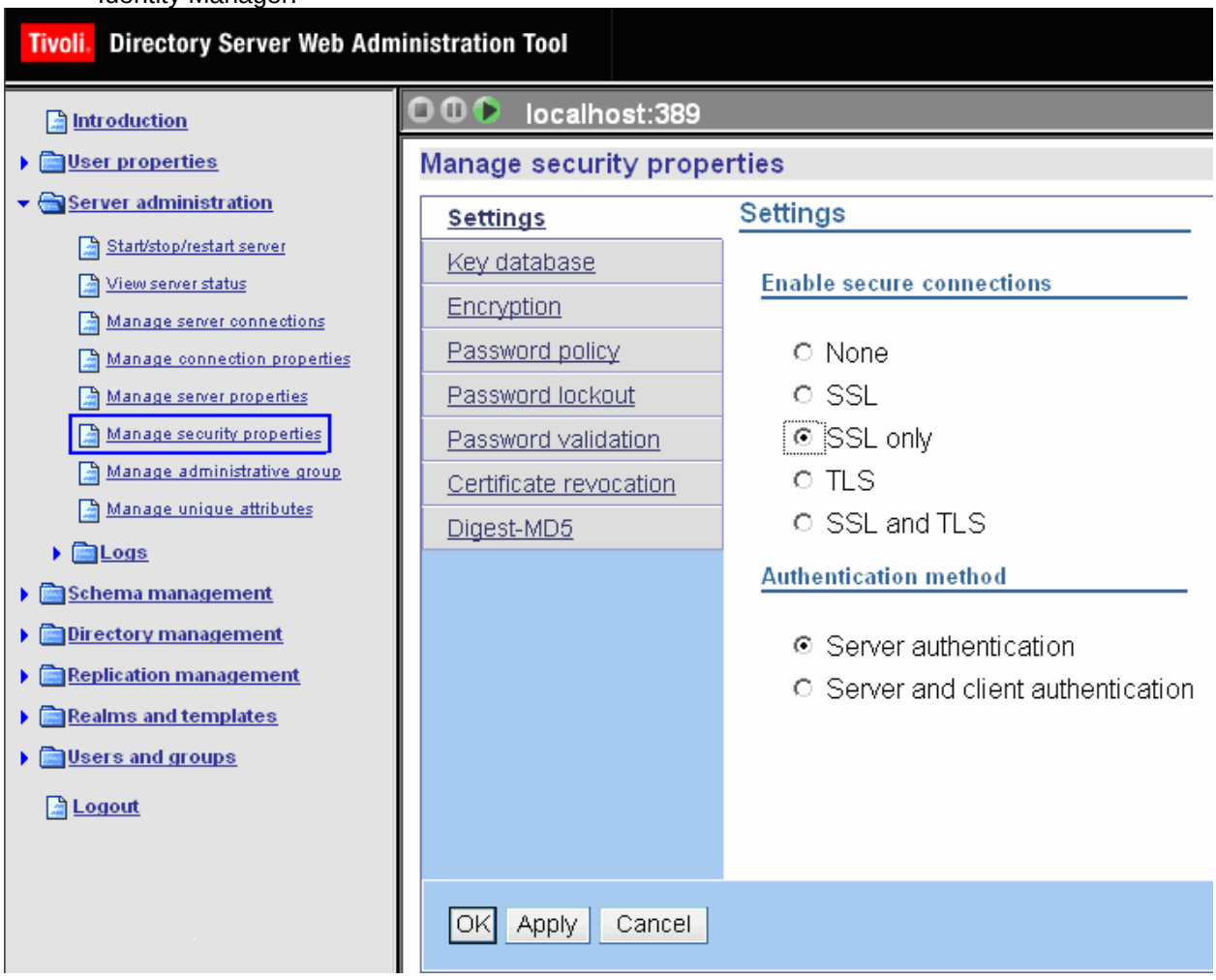
To configure SSL on IBM Tivoli Directory Server, complete these steps:

1. Log in to the Directory Server Web Administration Tool as the LDAP administrator.

2. Select **Server administration > Manage security properties > Settings** from the navigation tree displayed on the left panel.

3. In the **Enable secure connections** area, select **SSL only** to use only secure connections or select **SSL** to use both secure and non-secure connections.

4. In the **Authentication method property**, select **Server authentication**. This enables one-way SSL communication. Do not select **Server and client authentication**, because two-way SSL between the Tivoli Identity Manager Server and the LDAP Server is not supported by Tivoli Identity Manager.



Next, you need to define the path to the key database file, the password for the certificate (if you did not save it to a stash file, and the certificate label:

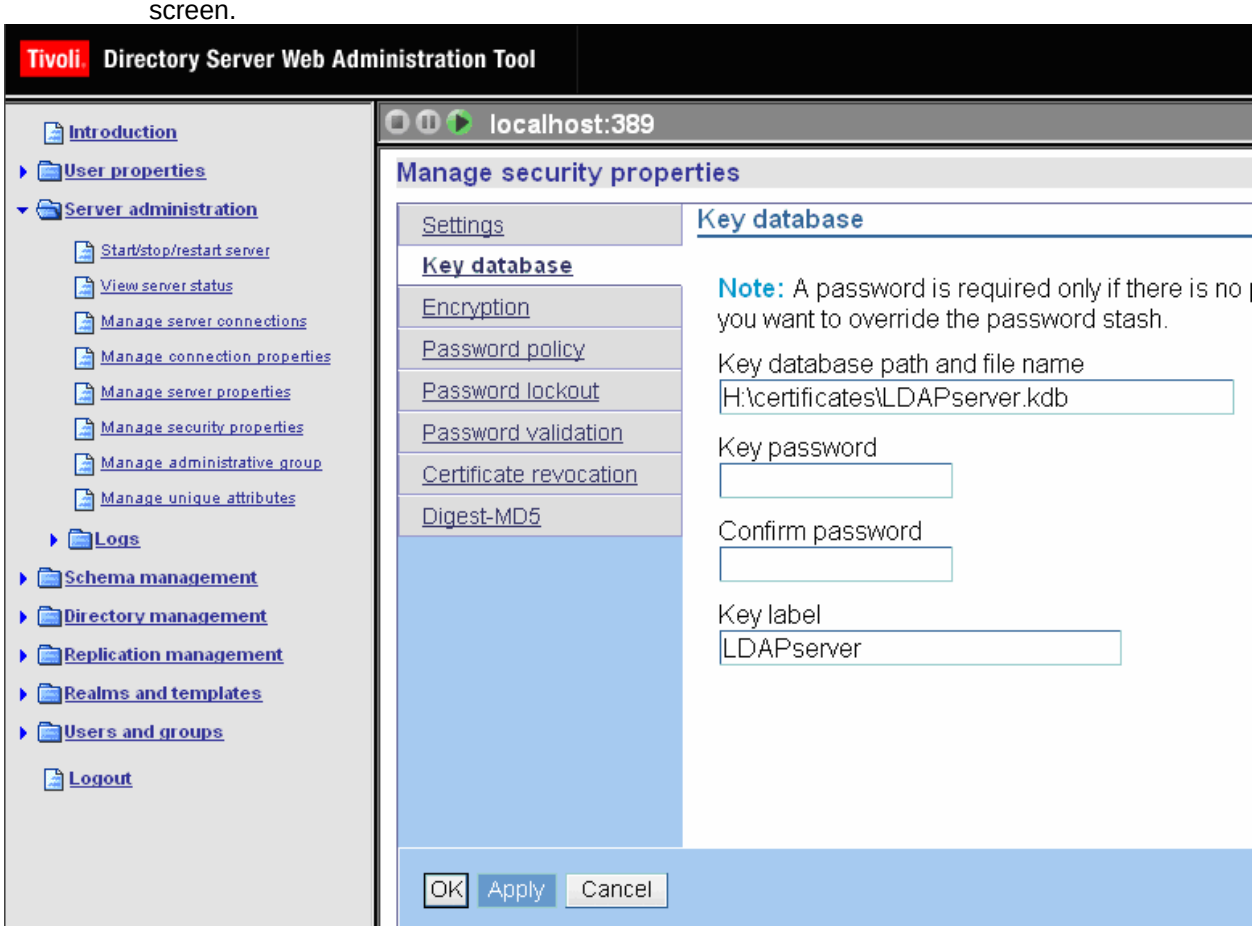
1. Select Server **administration > Manage security properties > Key database**.

2. In the **Key database path and file name** field, type the path for the key database file that you specified when you created the self-signed certificate. In Section 2 above, the example path was H:\certificates\.

3. In the **Password** field, type the password for the certificate file. If you saved the password in a stash file that is located in the same directory as the key database file, leave this field blank.

4. In the **Key label** field, type the label for the certificate that you specified in Section 2. In Section 2 above, the example was LDAPserver.

5. Apply the changes by clicking on **Apply** and then **OK** button present at the bottom of the screen.

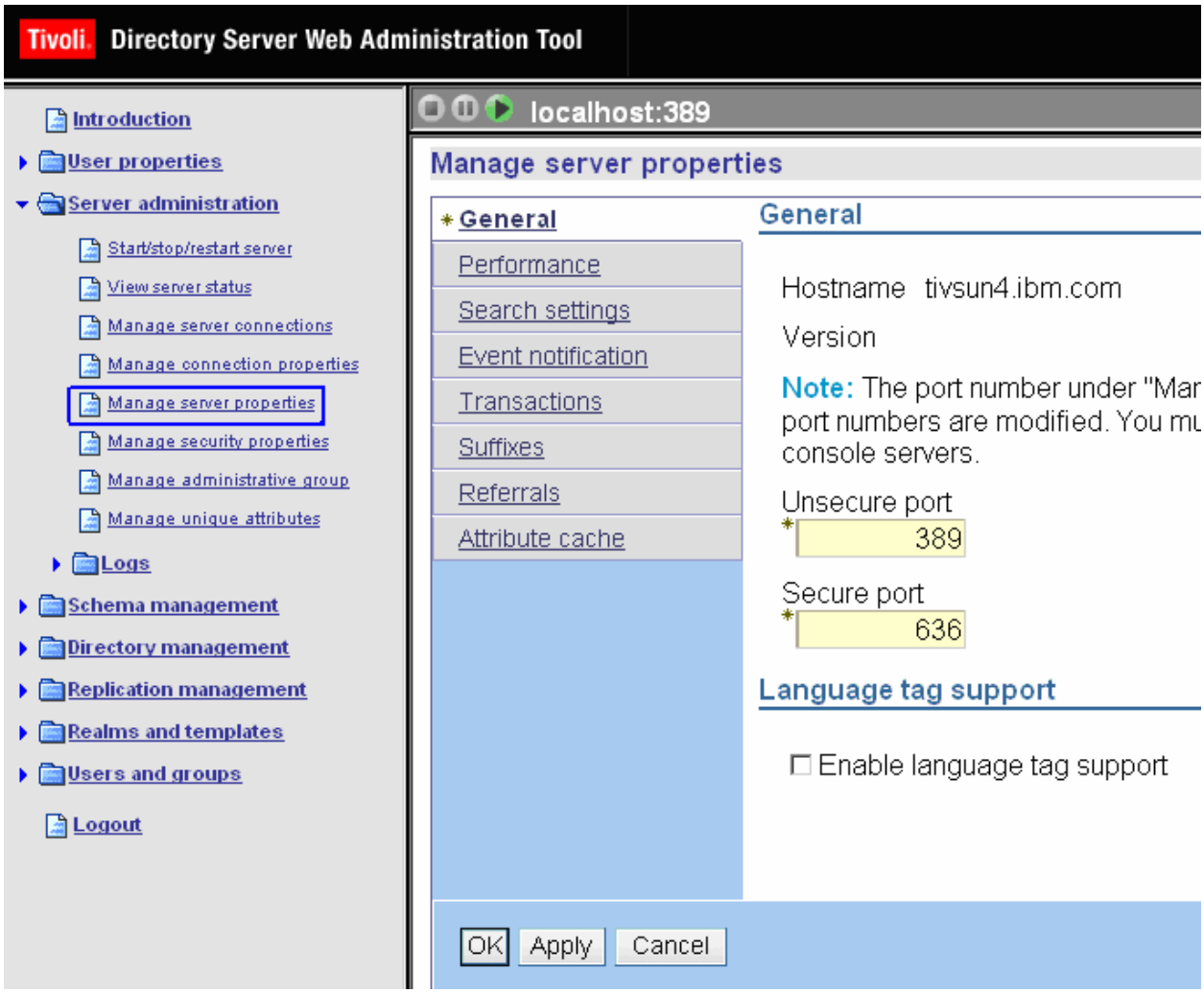


Next, you can optionally define a unique SSL port, if the default port of 636 is not available or is in use:

1. Select **Server administration > Manage server properties > General**.

2. In the **Secure port** field, type the port number that you want to use for SSL communication. The default is 636.

3. Apply the changes by clicking on **Apply** and then **OK** button present at the bottom of the screen.



Finally, you need to restart the directory server. View the ibmslapd.log file, and ensure that the server instance started successfully using SSL.

3.2 Configuring SSL for Sun ONE Directory Server

For detailed information on setting up SSL on the **Sun ONE Directory Server**, refer to the following Web site:

http://docs.sun.com/app/docs/prod/s1dirsrv

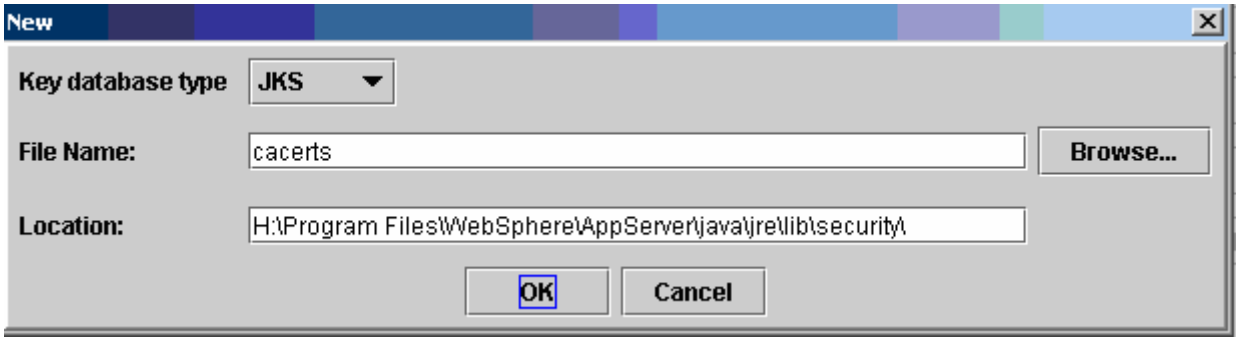
4 Configuring the SSL Connection between the Tivoli Identity Manager Server and the LDAP Server

Tivoli Identity Manager communicates with the LDAP Server using the JNDI APIs and LDAP. The secure communication – ldaps (LDAP over SSL) – uses JSSE, which uses a standard Java truststore and key store. To successfully configure the SSL connection between the Tivoli Identity Manager Server and the LDAP Server, you must import the self signed certificate (or CA certificate) created for the LDAP Server into the truststore that is used by JSSE (the IBM JSSE, which is part of WebSphere Application Server). Additionally, you must configure the Tivoli Identity Manager data service, which accesses the LDAP data to use SSL (configuring it to use the ldaps protocol instead of the ldap protocol) when communicating with the LDAP Server.

4.1 Installing the self signed certificate in the JSSE Truststore

In these instructions, the default truststore that is present in the JRE of the WebSphere Applicatoin Server is used. Also, the iKeyman utility is used to configure the certificates. To install the self signed certificate for the LDAP Server in the JSEE truststore, complete these steps:

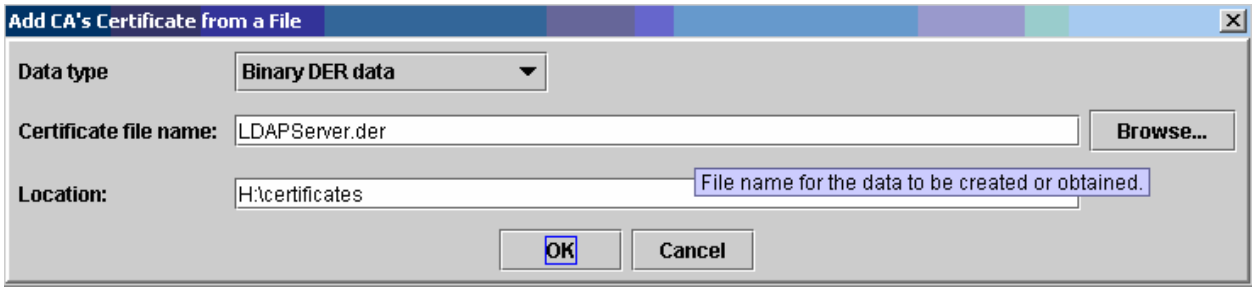
1. Start the ikeyman utility (ikeyman.bat or ikeyman.sh). It is located in the <WAS_HOME>/AppServer/bin directory.



2. From the **Key Database File** menu, select **New**.
3. In the File Name field, type cacerts. This is the default name for the JRE certificates file.
4. In the Location field, type <WAS_HOME>/AppServer/java/jre/lib/security/.
5. In the Password Prompt window, type the password for the keystore in the Password and Confirm Password window. The default password is `changeit`.
6. Click **OK**.

Next, you need to add the certificate extracted above into this certificate store.

1. In the main window, in the Key database content area, select Signer Certificates from the drop-down list.
2. Click **Add**.



3. From the **Data Type** drop-down list, select **Binary Der data**.
4. In the **Certificate file name** field, browse and locate the certificate file that was copied to the Tivoli Identity Manager server at the end of Section 2. In our example, it was the LDAPserver.der file. Verify that the appropriate directory is displayed in the **Location** field.

5. Click **OK**.

6. In the prompt that is displayed, type a label for this certificate. For example, type LDAPCA.

7. Click **OK**.

The certificate is added for the LDAP Server. You can now close the ikeyman utility.

4.2 Configuring Tivoli Identity Manager to Use SSL When Communicating with the LDAP Server

To configure Tivoli Identity Manager to use SSL when communicating with the LDAP server, complete these steps:

1. Edit the enRoleLDAPConnection.properties file in the <ITIM_HOME>\data directory.

2. Set the value of the **java.naming.security.protocol** property to **ssl**. This will indicate to IBM Tivoli Identity Manager server that it should communicate to LDAP using SSL.

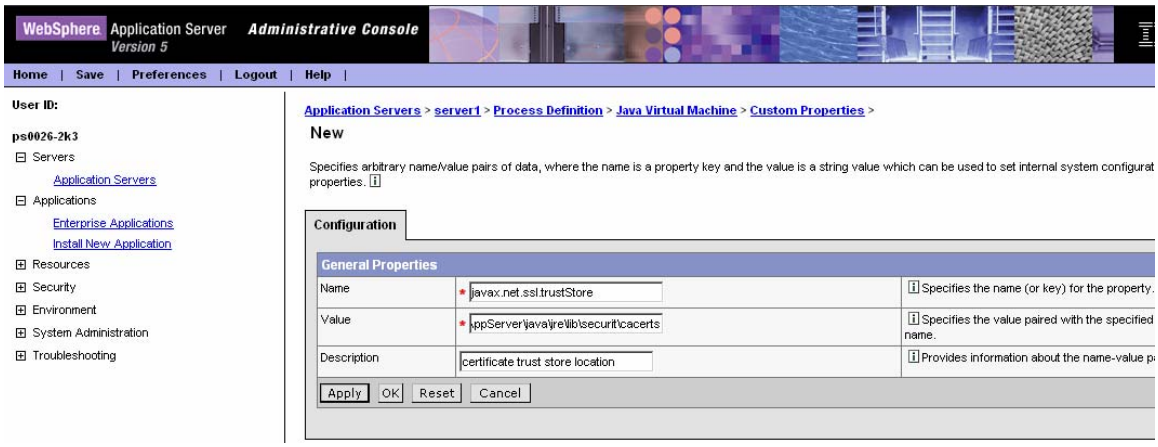
3. Set the port value on the **java.naming.provider.url** property to the SSL port number configured on directory server [LDAP]. For example, `java.naming.provider.url=ldap://localhost:636`.

4. Save the file. Close the editor.

4.3 Defining the Truststore and Password as a Custom Property on the JVM

Tivoli Identity Manager Server does not use the WebSphere Application Server **SSL Configuration Repositories** settings in the WebSphere Administrative Console **Security | SSL** tab. Instead, you must configure the SSL settings using the following menus to specify the javax properties. Complete these steps:

1. Select **Servers > Application Servers > <server name> Process Definition > Additional Properties > Java Virtual Machine > Custom Properties > New**.



2. Define the name of the javax properties that you have changed using the ikeyman key management tool. In section 4.1 we have shown how to install certificate into the trust store of the jvm used by WebSphere Application server. Alternately you can create your own certificate store location, in which case you will have to define some additional properties See Table 1 for the javax properties that you need to define.

Table 1: The javax properties that must be specified

Property name	Description	Default value
javax.net.ssl.trustStore	File path of the truststore file. You can use the truststore to install CA certificates and client certificates. If you do not use javax.net.ssl.keyStore to specify a client certificate, you must use this truststore.	*jre_install_dir*\lib\security\cacerts Example: c:\Program Files\WebSphere\AppServer\java\jre\lib\security\cacerts
javax.net.ssl.trustStorePassword	Password that protects the truststore.	changeit (This is the default password for the JVM default truststore.)
javax.net.ssl.keyStore	File path of the keystore file. The keystore contains the certificate that is used by the Tivoli Identity Manager Server. The certificate must be present either in the keystore or the truststore if the application operating as an SSL server (for example, an agent-based adapter) is set to require client authentication. If this property is not defined, the truststore must contain the certificate when client authentication is required.	None. The truststore file path is searched by default.
javax.net.ssl.keyStorePassword	Password that protects the keystore.	changeit (This is the default password for the JVM default keystore.)

4.4 Running the ldapConfig Utility and Verifying the Connection to the Directory Server and Running the Utilities That Access the LDAP Server

Complete this step *only* if you configured the LDAP Server to use secure connections only (specified SSL only in Section 2 above). You can temporarily configure the LDAP server to accept both secure SSL connections (ldaps) and non-secure connections (ldap), run these utilities, and then reconfigure the LDAP server to use secure connections only.

When you first install Tivoli Identity Manager, if the LDAP Server is configured to only use secure connections, the ldapConfig utility will not work during installation. You must skip the configuration during installation and run it again *after performing the steps given at the end of this section,* after the Tivoli Identity Manager installation has completed.

Same steps are to be performed in order to successfully run the following utilities present in <ITIM_HOME>/bin/<platform> directory:

- addindex
- addintegrity
- config_remote_services
- ldapClean
- remove_service_profiles
- serviceability

Complete these steps:

1. In the <ITIM_HOME>/data directory, edit the enRoleLDAPConnections.properties file.

2. Set the value of the **java.naming.security.protocol** property to **ssl**. This will indicate to IBM Tivoli Identity Manager server that it should communicate to LDAP using SSL.

3. ldapConfig, ldapUpgrade and runConfig use the jvm specified in **lax.nl.current.vm** property in the respective lax file. i.e. ldapConfig.lax, ldapUpgrade.lax and runConfig.lax. If you have installed the certificate in the certificate store of this jvm, you need not perform this step. Other wise you will have to specify additional properties given in Table 1 under section 4.3. The values of these properties are based on the location and password of the certificate store created by you. In this case add the additional properties at the end of <ITIM_HOME>\data\enRoleLDAPConnection.properties file.

   Say you have installed the certificate in a certificate file created at H:\\timCertificate.jks with the password secret. Perform the following steps.

   a. Edit the enRoleLDAPConnection.properties file in the <ITIM_HOME>\data directory.

   b. Add javax.net.ssl.trustStore= H:\\timCertificate.jks and javax.net.ssl.trustStorePassword=secret at the end of this file in separate lines.

   c. Save the file. Close the editor.

To run LoadDSMLSchema when SSL is enabled, perform the following steps:
1. Edit LoadDSMLSchema.cmd or LoadDSMLSchema.sh file depending on your platform.
2. Specify appropriate security certificate location and password as java command line options. Please refer the description of each of the four properties given in Table 1 under section 4.3. For example, to specify the trustStore location and password, modify the last line as follows:

```
"%JAVA_HOME%\bin\java" -Djavax.net.ssl.trustStore=H:\\timCertificate.jks -
Djavax.net.ssl.trustStorePassword=secret  -classpath "%CLASSPATH%"
com.ibm.dsml.LoadDSMLSchema %1 %2 %3 %4 %5
```